Bayesian Learning

Lecture series „Machine Learning“

Niels Landwehr

Research Group „Data Science“
Institute of Computer Science
University of Hildesheim

Agenda for Lecture

- Fundamental concepts of Bayesian learning
- Introductory example: coin tosses
- Bayesian linear regression

Agenda for Lecture

- Fundamental concepts of Bayesian learning
- Introductory example: coin tosses
- Bayesian linear regression

Review: Probabilistic Model for Data

- We are still concerned with supervised learning problems:
  - Models $f : \mathcal{X} \to \mathcal{Y}$
  - Training data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$

- Review: probabilistic assumptions about training data
  - Training instances independently drawn from joint distribution over inputs and outputs:

$$(\mathbf{x}_n, y_n) \sim p(\mathbf{x}, y)$$

  - According to product rule, can split up joint distribution into

$$\mathbf{x}_n \sim p(\mathbf{x})$$
$$y_n \sim p(y \mid \mathbf{x}_n)$$

- The instances $\mathbf{x}_n$ are sampled from a probability distribution over instances.
- $p(\mathbf{x})$ describes distribution over population of objects

- Given an instance $\mathbf{x}_n$, its label is drawn from a distribution $p(y \mid \mathbf{x}_n)$ that represents the relationship between input and output.

Review: Probabilistic Models

- Review: probabilistic models such as logistic regression define a conditional probability distribution

$$p(y \mid \mathbf{x}, \boldsymbol{\theta})$$

possible outputs — input — model

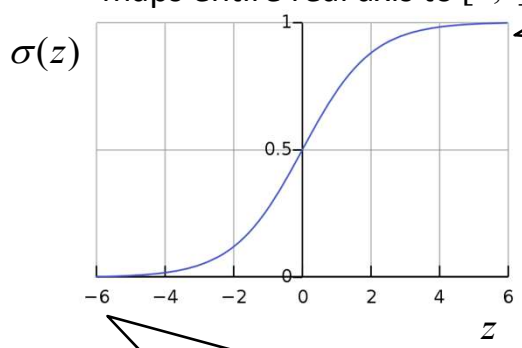
- For example, binary logistic regression: distribution over binary $y \in \{0,1\}$ by defining probability for positive class as sigmoid function of linear model

$$p(y=1 \mid \mathbf{x}, \boldsymbol{\theta}) = \sigma(\mathbf{x}^{\mathrm{T}}\boldsymbol{\theta})$$

$$p(y=0 \mid \mathbf{x}, \boldsymbol{\theta}) = 1 - \sigma(\mathbf{x}^{\mathrm{T}}\boldsymbol{\theta})$$

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

„Squashing function“ σ maps entire real axis to $[0,1]$



Probability approaches one for large positive values of linear model

Probability approaches zero for large negative values of linear model

Probabilistic Models: Maximum Likelihood

- So far, we have learned probabilistic models by maximum likelihood (lecture on linear classification):

Optimization Problem:

$$\boxed{\begin{aligned}\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \prod_{n=1}^{N} p(y_n \mid \mathbf{x}_n, \boldsymbol{\theta})\end{aligned}}$$

- This yields a single final model (parameter estimate) $\boldsymbol{\theta}^*$

- Question: can we really be sure that $\boldsymbol{\theta}^*$ is the best model?
  - The selection of the best model $\boldsymbol{\theta}^*$ out of all possible model parameters is based on a limited set of training data $\mathcal{D}$
  - Realistically, there is remaining uncertainty: even after having seen the training data, cannot be 100% sure what the best model is
  - **Idea: can we quantify this remaining uncertainty?**

Bayesian Model of Data Generation

- **Bayesian Learning**: extend the probabilistic model of how the training (and test) data are generated
- Specifically, reason probabilistically about models as well, not only data
- Statistically speaking, the model itself becomes a random variable

- **Bayesian model of how the data (and the true model) is generated**:
  1. The true model $\boldsymbol{\theta}^*$ is drawn from a so-called **prior distribution** $p(\boldsymbol{\theta})$ over possible models.
     - The true model $\boldsymbol{\theta}^*$ is unknown, but we make an assumption for the prior distribution $p(\boldsymbol{\theta})$
     - Assumption could be, for example, that models with small parameter values are more likely than models with large parameter values (idea of shrinkage)
  2. The inputs $\mathbf{x}_1,\ldots,\mathbf{x}_N$ are drawn independently from a distribution over inputs:
     $$\mathbf{x}_n \sim p(\mathbf{x})$$
  3. The outputs $y_1,\ldots,y_N$ are then drawn from the distribution given by the model $\boldsymbol{\theta}^*$
     $$y_n \sim p(y \mid \mathbf{x}, \boldsymbol{\theta}^*)$$

Bayes Rule and Posterior Probability of Model

- At the heart of Bayesian machine learning is the **Bayes rule**, a simple rule for any two random variables u, v which have a joint distribution $p(u,v)$:

$$p(u \mid v) = \frac{p(v \mid u)\, p(u)}{p(v)}$$

- In Bayesian learning, we apply this rule to models and data in the following way (rough idea, see below for a more formal/detailed treatment):

Bayes rule computes the so-called **posterior probability** of a model given the data: How likely is it that a particular model is the true model, given the data we have seen? The distribution tells us both about likely correct models and the remaining uncertainty.

This is the **likelihood**: which probability does the model assign to the training data? Can be computed given model and data.

This is the **prior**: what are a priori likely models (before seeing the data). Assumed known.

$$p(\text{Model} \mid \text{Data}) = \frac{p(\text{Data} \mid \text{Model})\, p(\text{Model})}{p(\text{Data})}$$

Normalizing factor that can in principle be computed from likelihood and prior

Bayes Rule and Posterior Probability of Model

- More formally, for a model of the form $p(y \mid \mathbf{x}, \boldsymbol{\theta})$, the Bayes rule reads as follows:

Models are given by parameter vectors

Likehood: probability of observed labels given input and model

Prior over model parameters

$$p(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{X})\, p(\boldsymbol{\theta})}{p(\mathbf{y} \mid \mathbf{X})}$$

Probability of data, only normalizing factor that is independent of model $\boldsymbol{\theta}$

- Here, $\mathbf{X}$ and $\mathbf{y}$ are the training data in the usual matrix/vector notation

- Proof: simply Bayes rule applied to the variables $\boldsymbol{\theta}, \mathbf{y}$ with joint distribution given by the conditional $p(\boldsymbol{\theta}, \mathbf{y} \mid \mathbf{X})$ (note that $p(\boldsymbol{\theta} \mid \mathbf{X}) = p(\boldsymbol{\theta})$ because prior is independent of input data $\mathbf{X}$)

- Note that because the model only describes a distribution $p(y \mid \mathbf{x}, \boldsymbol{\theta})$ over outputs given inputs and parameters, „Data“ in this case only refers to the observed labels $\mathbf{y}$

Bayesian Predictions

- Bayes rule gives us a posterior probability distribution over models given the data
- How do we obtain predictions for a novel test instance $\mathbf{x}_{new}$?

- **First approach**:
  - Find the most probable model given the prior and the data:

$$\begin{aligned}\boldsymbol{\theta}_{MAP} &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y}) \\ &= \arg\max_{\boldsymbol{\theta}} \frac{p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathbf{y} \mid \mathbf{X})} \\ &= \arg\max_{\boldsymbol{\theta}} p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})\end{aligned}$$

The model $\boldsymbol{\theta}_{MAP}$ is also called the **maximum a posteriori** (MAP) model

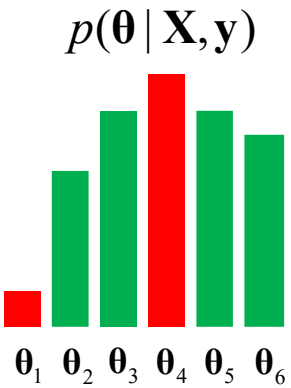
  - We can then compute a prediction $\hat{y}$ using this model:

$$\hat{y} = \arg\max_{y} p(y \mid \mathbf{x}_{new}, \boldsymbol{\theta}_{MAP})$$

Motivation: Bayesian Predictions

- The prediction with the maximum-a-posteriori model does not take into account the remaining uncertainty inherent in the posterior distribution

- Let's look at a toy example:
  - Assume that our model space only contains six models overall: $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \boldsymbol{\theta}_4, \boldsymbol{\theta}_5, \boldsymbol{\theta}_6\}$
  - Of those six models, two predict the negative class (red) and four predict the positive class (green)
  - Model $\boldsymbol{\theta}_4$ is the maximum a posteriori model, and would result in the prediction „negative class"
  - However, is this the right decision? There is a lot of remaining uncertainty, and most other models would predict the positive class...



- Would prefer to better take remaining uncertainty into account
- **Idea**: take into account predictions of **all models**, and weight them by their posterior probability (see next slide)

Bayesian Predictions

- **Second approach: Bayesian prediction**
- Idea: we do not limit ourselves to one model, but directly compute the most probable prediction given the evidence from the training data and the new instance:

Most probable prediction, given training data $\mathbf{X}, \mathbf{y}$ and the new instance $\mathbf{x}_{new}$

$$\hat{y} = \arg\max_y p(y \mid \mathbf{x}_{new}, \mathbf{X}, \mathbf{y})$$

Marginalization rule $p(a) = \int p(a,b)db$ applied to conditional distribution $p(y, \boldsymbol{\theta} \mid \mathbf{x}_{new}, \mathbf{X}, \mathbf{y})$

$$= \arg\max_y \int p(y, \boldsymbol{\theta} \mid \mathbf{x}_{new}, \mathbf{X}, \mathbf{y}) \, \mathrm{d}\boldsymbol{\theta}$$

Product rule $p(a,b) = p(a \mid b) p(b)$ applied to conditional distribution $p(y, \boldsymbol{\theta} \mid \mathbf{x}_{new}, \mathbf{X}, \mathbf{y})$

$$= \arg\max_y \int p(y \mid \boldsymbol{\theta}, \mathbf{x}_{new}, \mathbf{X}, \mathbf{y}) \, p(\boldsymbol{\theta} \mid \mathbf{x}_{new}, \mathbf{X}, \mathbf{y}) \, \mathrm{d}\boldsymbol{\theta}$$

$$= \arg\max_y \int p(y \mid \mathbf{x}_{new}, \boldsymbol{\theta}) \, p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y}) \, \mathrm{d}\boldsymbol{\theta}$$

Given the model, the training data does not influence the prediction:
$p(y \mid \boldsymbol{\theta}, \mathbf{x}_{new}, \mathbf{X}, \mathbf{y}) = p(y \mid \boldsymbol{\theta}, \mathbf{x}_{new})$

The learned model is independent of the new test input:
$p(\boldsymbol{\theta} \mid \mathbf{x}_{new}, \mathbf{X}, \mathbf{y}) = p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y})$

Bayesian Predictions

- **Bayesian prediction: average over the predictions of all models, weighted by their posterior probability**

$$\hat{y} = \arg\max_y \int p(y \mid \mathbf{x}_{new}, \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y})\, \mathrm{d}\boldsymbol{\theta}$$

Average over all models $\boldsymbol{\theta}$

Prediction of model $\boldsymbol{\theta}$

Posterior probability of model $\boldsymbol{\theta}$

- Bayesian prediction takes into account the predictive distributions of all models
- Models are weighted by their posterior probability

Agenda for Lecture

- Fundamental concepts of Bayesian learning
- Introductory example: coin tosses
- Bayesian linear regression

Example: Coin Tosses

- Introductory example for Bayesian learning: Tossing a binary coin *N* times

- Coin tossing experiment
  - Assume coin that can be tossed and then lands on head $(y=1)$ or tail $(y=0)$
  - The coin is not necessarily fair, and the probability of landing on head is $\theta \in [0,1]$:

$$p(y=1 \mid \theta) = \theta$$
$$p(y=0 \mid \theta) = 1-\theta$$

  - Assume we are tossing the coin *N* times, observing results $\mathbf{y} = (y_1, \ldots, y_N)^{\mathrm{T}}$ with $y_i \in \{0,1\}$
  - We can summarize the observations into counts for the outcomes head and tail:

$$N_h = \sum_{n=1}^{N} I(y_n = 1) \qquad \text{number of heads}$$

$$N_t = \sum_{n=1}^{N} I(y_n = 0) \qquad \text{number of tails}$$

Coin Tosses: Bayesian Parameter Estimates

- Coin tosses can be seen as a simple machine learning setting:
  - Unknown parameter θ is the model
  - Head and tail counts N_h, N_t are the training observations
  - Task is to learn model θ from training observations

- We want to follow a Bayesian approach:
  - What can we learn from the data about the model parameter θ ?
  - The data is given by the head and tail counts, and knowing N_h is sufficient given a fixed N
  - Bayes rule:

Posterior probability of a possible coin parameter θ after having seen the data

Likehood: probability of observed outcomes given a parameter θ

We need a prior over possible coin parameters θ

$$p(\theta \mid N_h) = \frac{p(N_h \mid \theta)\, p(\theta)}{p(N_h)}$$

Probability of data, only normalizing factor that is independent of model $\boldsymbol{\theta}$

Coin Tosses: Likelihood

- Likelihood $p(N_h \mid \theta)$: what is the probability of seeing exactly N_h heads (and therefore $N_t = N - N_h$ tails) for a given coin parameter θ ?

- Likelihood is given by the Binomial distribution:

$$\mathrm{Bin}(N_h \mid N, \theta) = \underbrace{\binom{N}{N_h}}_{\substack{\text{number of series} \\ \text{with } N_h \text{ heads}}} \underbrace{\theta^{N_h}(1-\theta)^{N-N_h}}_{\substack{\text{probability for a specific series of tosses} \\ \text{with } N_h \text{ heads and } N_t = N - N_h \text{ tails}}}$$

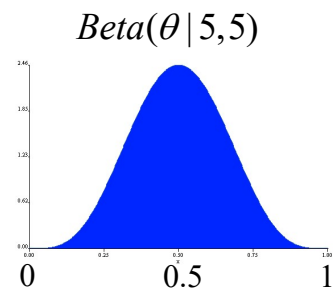
- "Binomial coefficient" $\binom{N}{N_h} = \dfrac{N!}{N_h!(N-N_h)!}$

Coin Tosses: Prior

- Prior $p(\theta)$: what is a good choice for a prior distribution over coin toss parameters θ ?

- Good choice for prior distribution is the so-called Beta distribution:

$$p(\theta) = Beta(\theta \mid \alpha_h, \alpha_t)$$
$$= \frac{\Gamma(\alpha_h + \alpha_t)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \theta^{\alpha_h - 1}(1-\theta)^{\alpha_t - 1}$$



- Here, $\Gamma : \mathbb{R} \to \mathbb{R}$ is a continuous extension of the factorial function:

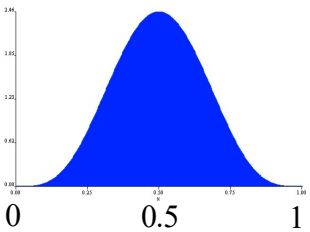
$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt \qquad \forall n \in \mathbb{N} : \Gamma(n) = (n-1)!$$

- The values α_h, α_t are parameters of the Beta distribution that determine the shape of the prior. Called hyperparameters to separate them from model parameter θ

Coin Tosses: Prior

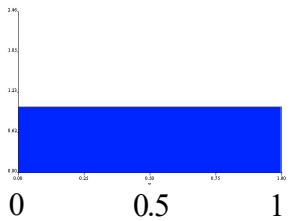
- Examples for Beta distributions with different hyperparameters:

$\alpha_h = 5, \quad \alpha_t = 5$



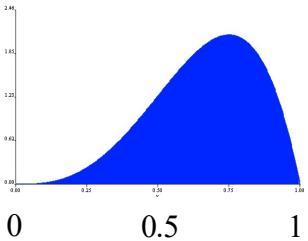
Prior assumption:
Most probably the coin is approximately fair, unlikely to always show head or always show tail

$\alpha_h = 1, \quad \alpha_t = 1$



Prior assumption:
No prior assumptions about parameter, so-called uninformative prior

$\alpha_h = 4, \quad \alpha_t = 2$



Prior assumption:
Coin will probably be biased towards landing head

Why a Beta Prior?

- Why did we choose this prior distribution?
- Structural similarity to likelihood function:

Prior: $$p(\theta) = Beta(\theta \mid \alpha_h, \alpha_t) = \frac{\Gamma(\alpha_h + \alpha_t)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \theta^{\alpha_h - 1}(1-\theta)^{\alpha_t - 1}$$

Likelihood: $$p(N_h \mid \theta) = Bin(N_h \mid N, \theta) = \binom{N}{N_h} \theta^{N_h}(1-\theta)^{N_t}$$

- This makes it easier to compute the posterior distribution over parameters given the empirical data (number of heads): according to Bayes rule,

$$p(\theta \mid N_h) = \frac{1}{Z} p(N_h \mid \theta) p(\theta)$$

Prior over possible coin parameters θ

Posterior probability of a possible coin parameter θ after having seen the data

Normalizer

Likehood: probability of observed outcomes given a parameter θ

Posterior is Again Beta Distribution

- Let‘s compute the posterior using Bayes rule:

$$
\begin{aligned}
p(\theta \mid N_h) &= \frac{p(N_h \mid \theta)\, p(\theta)}{p(N_h)} \\
&= \frac{1}{Z} \mathrm{Bin}(N_h \mid N, \theta)\, Beta(\theta \mid \alpha_h, \alpha_t) \\
&= \frac{1}{Z} \binom{N}{N_h} \theta^{N_h} (1-\theta)^{N_t} \frac{\Gamma(\alpha_h + \alpha_t)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \theta^{\alpha_h - 1} (1-\theta)^{\alpha_t - 1} \\
&= \frac{1}{Z'} \theta^{\alpha_h + N_h - 1} (1-\theta)^{\alpha_t + N_t - 1}
\end{aligned}
$$

What is *Z‘*?
What is the correct normalizer such that the final distribution integrates to one?

Posterior is Again Beta Distribution

- Let‘s compute the posterior using Bayes rule:

$$
\begin{aligned}
p(\theta \mid N_h) &= \frac{p(N_h \mid \theta)\, p(\theta)}{p(N_h)} \\
&= \frac{1}{Z} \mathrm{Bin}(N_h \mid N, \theta)\, Beta(\theta \mid \alpha_h, \alpha_t) \\
&= \frac{1}{Z} \binom{N}{N_h} \theta^{N_h} (1-\theta)^{N_t} \frac{\Gamma(\alpha_h + \alpha_t)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \theta^{\alpha_h - 1} (1-\theta)^{\alpha_t - 1} \\
&= \frac{1}{Z'} \theta^{\alpha_h + N_h - 1} (1-\theta)^{\alpha_t + N_t - 1} \\
&= \frac{\Gamma(\alpha_h + N_h + \alpha_t + N_t)}{\Gamma(\alpha_h + N_h)\Gamma(\alpha_t + N_t)} \theta^{\alpha_h + N_h - 1} (1-\theta)^{\alpha_t + N_t - 1} \\
&= Beta(\theta \mid \alpha_h + N_h, \alpha_t + N_t)
\end{aligned}
$$

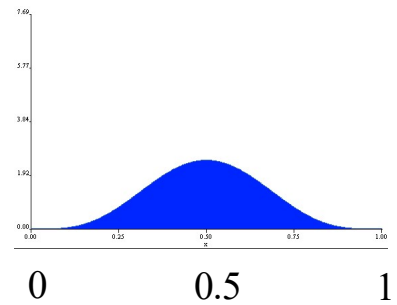
Has to be this normalizer, because this is the normalizer of the Beta distribution with hyperparameters $\alpha_h + N_h,\ \alpha_t + N_t$

- Posterior is again a Beta distribution, with new hyperparameters $\alpha_h + N_h,\ \alpha_t + N_t$
- **We call this a conjugate prior: posterior is in the same distribution family as prior**

Example: Posterior For Coin Tosses

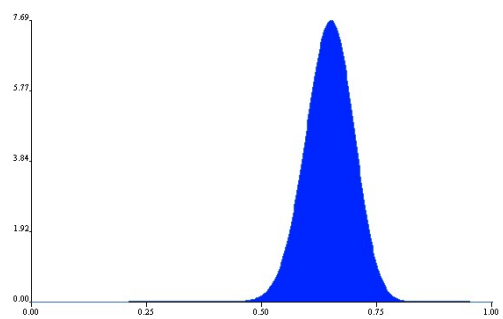
- Example: Computing posterior for coin tosses
- Let‘s say we choose a prior distribution $p(\theta) = Beta(\theta \mid 5,5)$

Prior expresses the belief that fair coins are more probable than unfair coins, and very unfair coins (θ near zero or one) are very improbable



- Let‘s assume we have thrown the coin $N = 75$ times, and have observed $N_h = 50$ times „Head“ and $N_t = 25$ times „Tail“
- According to posterior calculation on last slide, the posterior is again a Beta distribution with new hyperparameters 5+50, 5+25: $Beta(\theta \mid 55,30)$

Posterior belief: coin is probably unfair ($\theta > 0.5$). There is some remaining uncertainty about how unfair the coin is



Coin Tosses: Bayesian Prediction

- Bayesian prediction for coin tosses:
  - Assume we have seen the result of *N* coin tosses in the training data
  - What is the probability for a new toss of the same coin to land on „Head“ or „Tail“?

- Bayesian approach: do not use a single model to predict, but average over models weighted by posterior

N observed tosses (training data)

New coin toss

$$p(y_{new}=1\,|\,\mathbf{y}) = \int p(y_{new}=1\,|\,\theta)\,p(\theta\,|\,\mathbf{y})\,d\theta$$

$$= \int \theta\, Beta(\theta\,|\,\alpha_h + N_h, \alpha_t + N_t)\,d\theta$$

$p(y_{new}=1\,|\,\theta)=\theta$, plug in posterior

$$= \frac{\alpha_h + N_h}{\alpha_h + \alpha_t + N_h + N_t}$$

Expectation of Beta distribution with hyperparameters $\alpha_h + N_h, \alpha_t + N_t$. Standard result, no proof here.

- Bayesian prediction for the example of last slide: probability for next coin toss to land on head would be $55/85 \approx 0.647$

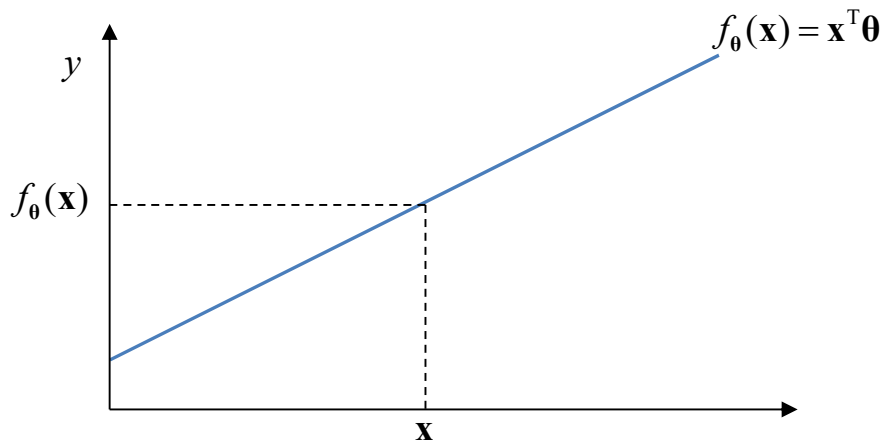
Agenda for Lecture

- Fundamental concepts of Bayesian learning
- Introductory example: coin tosses
- Bayesian linear regression

Review: Linear Regression

- Review: linear regression model
  - Defines model $f_{\boldsymbol{\theta}} : \mathbb{R}^M \to \mathbb{R}$
  - Prediction is linear function of input attributes:

$$
\begin{aligned}
f_{\boldsymbol{\theta}}(\mathbf{x}) &= \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_M x_M \\
&= \mathbf{x}^{\mathrm{T}} \boldsymbol{\theta}
\end{aligned}
$$



Linear Regression as a Probabilistic Model

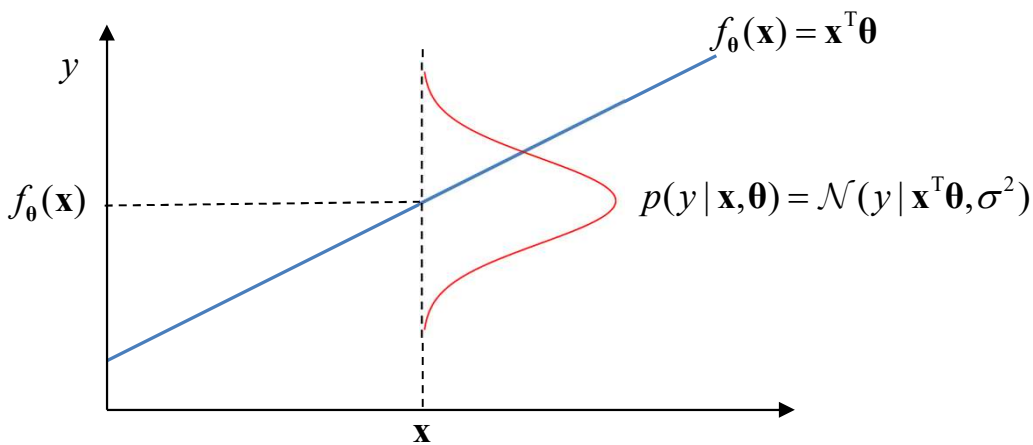
- Linear regression can be extended to a probabilistic model as follows

- Define a distribution over the output $y \in \mathbb{R}$ given the input $\mathbf{x} \in \mathbb{R}^M$:

$$p(y \mid \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y \mid \mathbf{x}^{\mathrm{T}}\boldsymbol{\theta}, \sigma^2)$$

Normal distribution with mean $\mathbf{x}^{\mathrm{T}}\boldsymbol{\theta}$ and variance σ^2.
Variance σ^2 is a hyperparameter of model

- Encodes some uncertainty about prediction



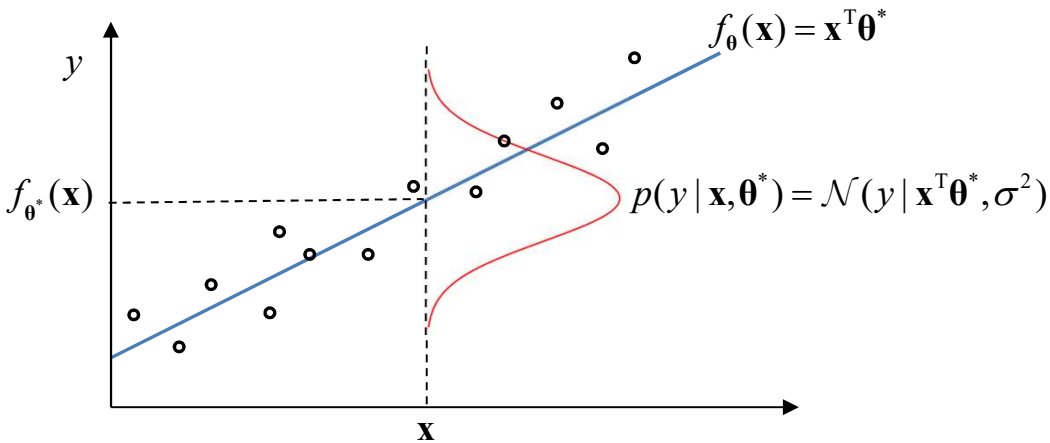
Linear Regression as a Probabilistic Model

- Remember that in a Bayesian setting, we assume that the data (labels) have been generated from an unknown true model $\boldsymbol{\theta}^*$:

$$y_n \sim p(y \mid \mathbf{x}, \boldsymbol{\theta}^*)$$

- In this case, this means that we assume that the labels are generated from a linear model plus Gaussian noise:

$$y_n = \mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}^* + \epsilon_n \qquad \epsilon_n \sim \mathcal{N}(\epsilon \mid 0, \sigma^2)$$



Bayes Rule for Probabilistic Linear Regression

- We use Bayes rule to compute the posterior probability:

Posterior distribution over model parameters

Likehood: probability of observed labels given inputs and model

We need a prior over model parameters

$$p(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{X})\, p(\boldsymbol{\theta})}{p(\mathbf{y} \mid \mathbf{X})}$$

Probability of data, only normalizing factor that is independent of model $\boldsymbol{\theta}$

- We need to
  - compute likelihood
  - define prior distribution
  - derive posterior distribution

Likelihood for Probabilistic Linear Regression

- Likelihood for probabilistic regression model:

$$p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) = p(y_1, \ldots, y_N \mid \mathbf{x}_1, \ldots, \mathbf{x}_N, \boldsymbol{\theta})$$

$$= \prod_{n=1}^{N} p(y_n \mid \mathbf{x}_n, \boldsymbol{\theta})$$

Independent training instances

$$= \prod_{n=1}^{N} \mathcal{N}(y_n \mid \mathbf{x}_n^{\mathrm{T}} \boldsymbol{\theta}, \sigma^2)$$

Plugging in model

$$= \mathcal{N}(\mathbf{y} \mid \mathbf{X}^{\mathrm{T}} \boldsymbol{\theta}, \sigma^2 \mathbf{I})$$

Product of univariate normal distributions can be written as multivariate normal distribution:

- Mean vector is vector of predictions:

$$\mathbf{X}^{\mathrm{T}} \boldsymbol{\theta} = \begin{pmatrix} \mathbf{x}_1{}^T \boldsymbol{\theta} \\ \ldots \\ \mathbf{x}_N{}^T \boldsymbol{\theta} \end{pmatrix}$$

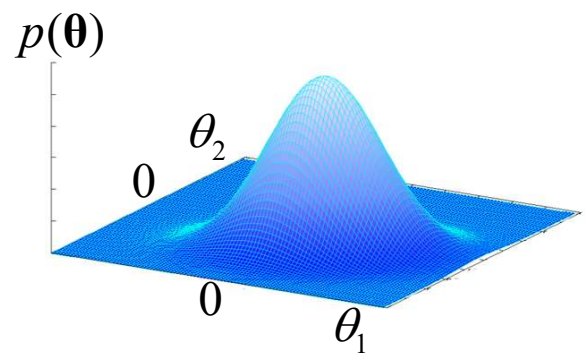
- Covariance matrix is a diagonal matrix $\sigma^2 \mathbf{I}$

Prior for Probabilistic Linear Regression

- Prior for probabilistic regression model:
  - We need to define a prior over parameter vectors $\boldsymbol{\theta} \in \mathbb{R}^M$
  - Similar idea as for regularization/shrinkage: we prefer parameter values that are not too large (in absolute value)

- As prior distribution, choose a multivariate normal distribution with mean zero and a diagonal covariance matrix:

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \mid \mathbf{0}, \sigma_p^2 \mathbf{I})$$



- Hyperparameter σ_p^2 controls variance of prior distribution and thereby strength of preference for small parameters
  - small σ_p^2 : low variance, very „peaked“ distribution, strong regularization
  - large σ_p^2 : high variance, flat distribution, less regularization

Posterior for Probabilistic Linear Regression

- The prior distribution is a conjugate prior for the likelihood computed above: the product of prior and likelihood is again a multivariate normal distribution

$$p(\boldsymbol{\theta} \mid \mathbf{y}, \mathbf{X}) = \frac{1}{Z} p(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{X}) p(\boldsymbol{\theta}) \qquad \text{Bayes rule}$$

$$= \frac{1}{Z} \mathcal{N}(\mathbf{y} \mid \mathbf{X}^{\mathrm{T}}\boldsymbol{\theta}, \sigma^2 \mathbf{I}) \cdot \mathcal{N}(\boldsymbol{\theta} \mid \mathbf{0}, \sigma_p^2 \mathbf{I}) \qquad \text{plugging in likelihood and prior}$$

$$= \mathcal{N}(\boldsymbol{\theta} \mid \overline{\boldsymbol{\theta}}, \mathbf{A}^{-1}) \qquad \text{compute product of normal terms - no details here}$$

$$\text{where } \mathbf{A} = \sigma^{-2}\mathbf{X}^{\mathrm{T}}\mathbf{X} + \sigma_p^{-2}\mathbf{I} \quad \text{and} \quad \overline{\boldsymbol{\theta}} = \sigma^{-2}\mathbf{A}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

- Posterior is again normally distributed, with a new mean vector given by $\overline{\boldsymbol{\theta}}$ and a new covariance matrix $\mathbf{A}^{-1}$

Posterior for Probabilistic Linear Regression

- The maximum a posteriori model is

$$\begin{aligned}\boldsymbol{\theta}_{MAP} &= \arg\max_{\boldsymbol{\theta}} \mathcal{N}(\boldsymbol{\theta} \mid \overline{\boldsymbol{\theta}}, \mathbf{A}^{-1}) \\ &= \overline{\boldsymbol{\theta}}\end{aligned}$$

- Note the similarity to the solution of ridge regression:
  - MAP solution:

$$\boldsymbol{\theta}_{MAP} = \sigma^{-2}(\sigma^{-2}\mathbf{X}^{\mathrm{T}}\mathbf{X} + \sigma_p^{-2}\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

  - Ridge regression (squared loss, L2-regularization):

$$\boldsymbol{\theta}^* = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + N\lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$

- For appropriately chosen hyperparameters, the solution is the same