02477 – Bayesian Machine Learning: Lecture 7

Michael Riis Andersen

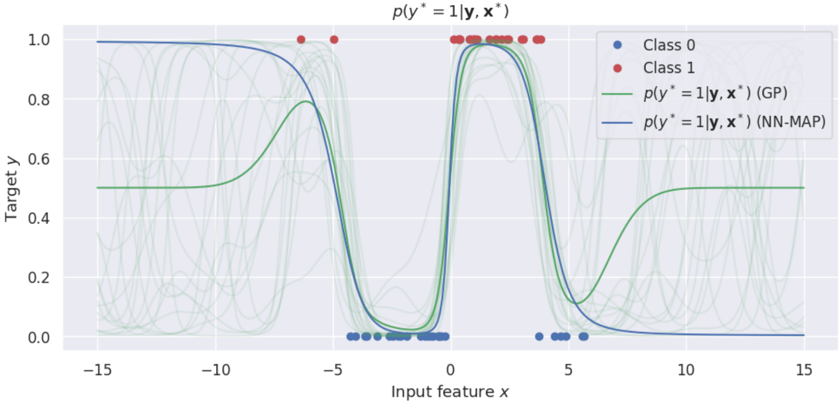
Technical University of Denmark,
DTU Compute, Department of Applied Math and Computer Science

Outline

1. Gaussian processes and neural networks
2. Generalized linear models and non-Gaussian likelihoods
3. Generalization and evaluation
4. Decision theory
5. Calibration

Gaussian processes and neural networks

From last week's exercise



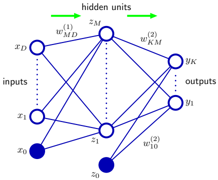
- Some NN researchers are striving to make NNs behave more like Gaussian processes
- Some GP researchers are striving to make GPs flexible and as easy to scale as NNs
- Exploring relationships between GPs and NNs

Infinitely wide neural networks I

Exploring connections between GPs and NNs

- Consider a network with a single hidden layer with H neurons and one output

$$\boldsymbol{z}(\boldsymbol{x}) = h(\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}_1)$$
$$f(\boldsymbol{x}) = \boldsymbol{W}_2\boldsymbol{z}(\boldsymbol{x}) + \boldsymbol{b}_2$$

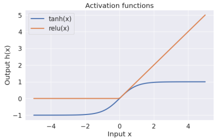


- For $\boldsymbol{x} \in \mathbb{R}^D$, then $\boldsymbol{W}_1 \in \mathbb{R}^{H\times D}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{H\times 1}$

- *Universal approximators* for many classes of functions

- Let's now make some assumptions

  1. Activation function h is bounded, e.g. $h(x) = \tanh(x)$
  2. $\boldsymbol{W}_1$ and $\boldsymbol{b}_1$ have i.i.d. zero-mean Gaussian priors
  3. $\boldsymbol{W}_2$ and $\boldsymbol{b}_2$ have zero-mean and prior variances σ_w^2 and σ_b^2
  4. Prior variance of $\sigma_w^2 = \frac{1}{H}$



- What is the mean and variance of $f(\boldsymbol{x})$?

$$\mathbb{E}\left[f(\boldsymbol{x})\right] = \mathbb{E}\left[\boldsymbol{W}_2\boldsymbol{z}(\boldsymbol{x}) + \boldsymbol{b}_2\right] = \sum_{j=1}^{H} \mathbb{E}\left[w_j\right]\mathbb{E}\left[h_j(\boldsymbol{x})\right] + \mathbb{E}\left[\boldsymbol{b}_2\right] = 0$$

Infinitely wide neural networks II

- The neural network is a zero-mean stochastic process, i.e.

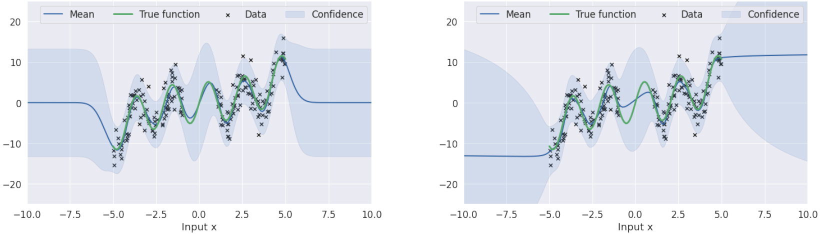
$$\mathbb{E}\left[f(\boldsymbol{x})\right] = 0$$

- Let number of neurons H go to infinity, i.e. $H \to \infty$

- CLT implies the neural network $f(\boldsymbol{x})$ converges to a Gaussian process

$$k_{\mathrm{NN}}(\boldsymbol{x}, \boldsymbol{x}') \equiv \frac{2}{\pi} \sin^{-1} \frac{2\tilde{\boldsymbol{x}}^T \tilde{\boldsymbol{x}}'}{\sqrt{(1 + 2\tilde{\boldsymbol{x}}^T \tilde{\boldsymbol{x}})(1 + 2\tilde{\boldsymbol{x}}'^T \tilde{\boldsymbol{x}}')}} \qquad \text{for} \qquad \tilde{\boldsymbol{x}} = [1, \boldsymbol{x}]^T$$

- Comparing *squared exponential* and *neural network kernels*



Generalized linear models and non-Gaussian likelihoods

Likelihoods as observation models: Why bother?

- Consider a linear model

$$f(\boldsymbol{x}) = \phi(\boldsymbol{x})^T \boldsymbol{w}$$

- Regression using Gaussian likelihood

$$y_n | f_n \sim \mathcal{N}(y_n | f_n, \beta^{-1})$$

- Binary classification with sigmoid function $\sigma(\cdot)$

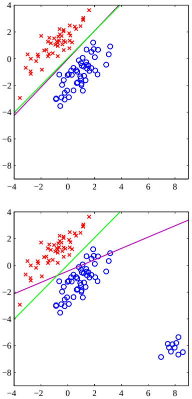
$$y_n | f_n \sim \mathrm{Ber}(y_n | \sigma(f_n))$$

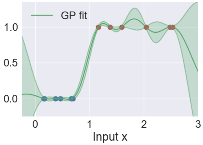
- ..., but why not just use regression for everything?

- More general setting: *generalized linear models* GLMs

$$y_n | f_n \sim p(y_n | f_n)$$

Least squared vs logistic regression

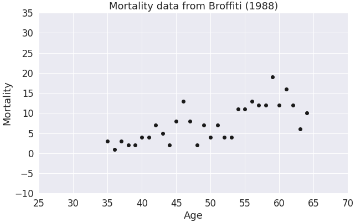


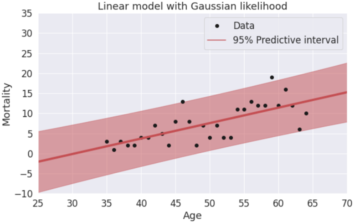


Common likelihoods in machine learning

Likelihood	Support	Example
Gaussian	$y_n \in \mathbb{R}$	Standard regression
Student's t	$y_n \in \mathbb{R}$	Standard regression with heavier tails
Exponential	$y_n \in (0, \infty)$	Strictly positive target values
Gamma	$y_n \in (0, \infty)$	Strictly positive target values
Bernoulli	$y_n \in \{0, 1\}$	Binary classification
Binomial	$y_n \in \{0, 1, \ldots, N\}$	Sequence of Bernoulli trials
Poisson	$y_n \in \{0, 1, 2, 3, \ldots, \}$	Count data
Categorial	$y_n \in \{0, 1, 2, \ldots, K\}$	Multi-class classification (requires K latent fucntions)

Example: analyzing mortality as a function of age





Under this model, the posterior predictive distribution for age $= 25$ is $\mathcal{N}(-2.61, 14.76)$ meaning that $p(\text{mortality} < 0|\text{age} = 25) \approx 0.75$

Common likelihoods in machine learning

Likelihood	**Support**	**Example**
Gaussian	$y_n \in \mathbb{R}$	Standard regression
Student's t	$y_n \in \mathbb{R}$	Standard regression with heavier tails
Exponential	$y_n \in (0, \infty)$	Strictly positive target values
Gamma	$y_n \in (0, \infty)$	Strictly positive target values
Bernoulli	$y_n \in \{0, 1\}$	Binary classification
Binomial	$y_n \in \{0, 1, \ldots, N\}$	Sequence of Bernoulli trials
Poisson	$y_n \in \{0, 1, 2, 3, \ldots, \}$	Count data
Categorial	$y_n \in \{0, 1, 2, \ldots, K\}$	Multi-class classification (requires K latent fucntions)

Spend 5 minutes on the DTU Learn quiz: "Lecture 7a: Common likelihoods in machine learning."

Discuss what type of observation model might be appropriate for ...

1. Predicting whether a student will pass a course (based on the student's previous grades)
2. Predicting the time it takes to finish a written exam
3. Predicting number of errors in a multiple choice exam
4. Predicting whether the student will fill out the exam with a blue, red or black pen

Generalized linear models in three steps

The components of a generalized linear model (GLM)

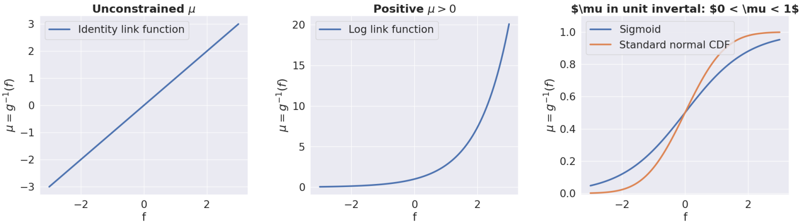
1. The linear model

$$f(\boldsymbol{x}) = \phi(\boldsymbol{x})^T \boldsymbol{w}$$

2. The *link function* g that relates the mean of the linear model to the mean of the response variable $\mathbb{E}[y(\boldsymbol{x})|\boldsymbol{x}] = \mu(\boldsymbol{x})$

$$g(\mu(\boldsymbol{x})) = f(\boldsymbol{x}) \qquad \iff \qquad \mathbb{E}[y|\boldsymbol{x}] = \mu(\boldsymbol{x}) = g^{-1}[f(\boldsymbol{x})]$$

3. The distribution $p(y_n|\boldsymbol{x}_n)$ for the response variable y_n, e.g. Poisson, binomial, gamma etc.



Example: Bayesian Poisson Regression I

- Step 1: We assume a linear model with $\boldsymbol{x} = [1 \quad \text{age}]^T$

$$f(\text{age}) = w_0 + w_1 \text{age} \qquad \Longleftrightarrow \qquad f(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{w}$$

- Step 2: Since $\mu > 0$, we use the log link function $\log(\mu) = f$ or equivalently

$$\mu(\boldsymbol{x}) = \exp(f(\boldsymbol{x})) = \exp(\boldsymbol{x}^T \boldsymbol{w})$$

- Step 3: We use a Poisson likelihood for count data

$$\text{Poisson}(y_n = k|\mu) = \frac{\mu^k \exp(-\mu)}{k!}$$

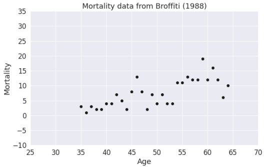
where $\mu > 0$ is the mean parameter.

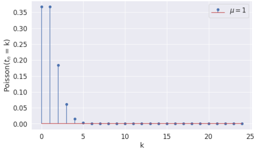
- Thus, the likelihood becomes

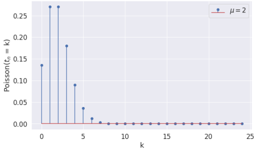
$$y_n | x_n, \boldsymbol{w} \sim \text{Poisson}(\mu_n)$$

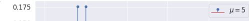
- We use a Gaussian prior for $\boldsymbol{w}$

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \boldsymbol{I})$$









Example: Bayesian Poisson Regression II

- Using $\mu_n = \mu(\boldsymbol{x}) = \exp(\boldsymbol{x}_n^T \boldsymbol{w})$, the joint model becomes

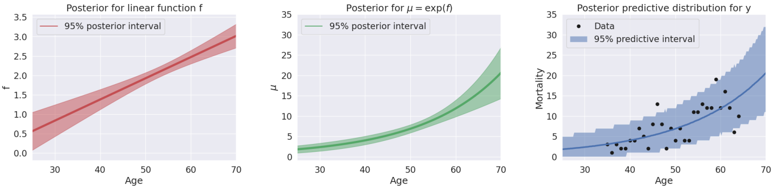
$$p(\boldsymbol{y}, \boldsymbol{w}) = p(\boldsymbol{y}|\boldsymbol{w})p(\boldsymbol{w}) = \prod_{n=1}^{N} \text{Poisson}(y_n|\mu_n)\mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \boldsymbol{I})$$

- The posterior is again intractable, so we use a Laplace approximation again

$$p(\boldsymbol{w}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{y})} \approx q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}, \boldsymbol{S})$$

- The (approximate) posterior predictive distribution for a new point $\boldsymbol{x}_*$ with $\mu_* = g^{-1}(\boldsymbol{x}_*^T \boldsymbol{w})$

$$p(y_* = k|\boldsymbol{y}) \approx \int p(y_* = k|\boldsymbol{w})q(\boldsymbol{w})\mathrm{d}\boldsymbol{w} = \int \text{Poisson}(y_* = k|\mu_*)\mathcal{N}(\boldsymbol{w}|\boldsymbol{m}, \boldsymbol{S})\mathrm{d}\boldsymbol{w}$$



Example: Bayesian Poisson Regression III

- The (approximate) posterior predictive distribution for a new point $\boldsymbol{x}_*$ with $\mu_* = g^{-1}(\boldsymbol{x}_*^T \boldsymbol{w})$

$$p(y_* = k|\boldsymbol{y}) \approx \int \text{Poisson}(y_* = k|\mu_*)\mathcal{N}(\boldsymbol{w}|\boldsymbol{m}, \boldsymbol{S})\text{d}\boldsymbol{w}$$

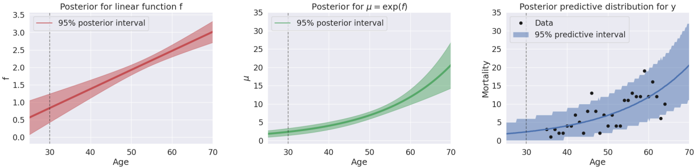
- Calculating predictions for age = 30. Let $\boldsymbol{x}_* = [1 \quad 30]$, then $f_* = \boldsymbol{x}_*^T \boldsymbol{w}$

$$p(f_*|\boldsymbol{y}) = \mathcal{N}(f_*|\boldsymbol{x}_*^T \boldsymbol{m}, \boldsymbol{x}_*^T \boldsymbol{S} \boldsymbol{x}_*) \approx \mathcal{N}(f_*|0.8, 0.2^2)$$

- We can calculate the distributions of $\mu_*|\boldsymbol{y}$ and $y_*|\boldsymbol{y}$ using samples $f_*^{(s)} \sim \mathcal{N}(f_*|0.8, 0.2^2)$ for $s = 1, \dots, S$. For each sample $f_*^{(s)}$, ...
  1. Compute $\mu_*^{(s)} = \exp(f_*^{(s)})$
  2. Sample $y_*^{(s)}|\mu_*^{(s)} \sim \text{Poisson}(\mu_*^{(s)})$

- Finally, we calculate the sample means (or variances, percentiles etc)

$$\mathbb{E}\left[\mu_*^{(s)}|\boldsymbol{y}\right] \approx \frac{1}{S}\sum_{s=1}^{S} \mu_*^{(s)} = 2.34 \qquad \mathbb{E}\left[y_*^{(s)}|\boldsymbol{y}\right] \approx \frac{1}{S}\sum_{s=1}^{S} y_*^{(s)} = 2.34$$



Generalized GP/NN models in three steps

The components of a generalized model

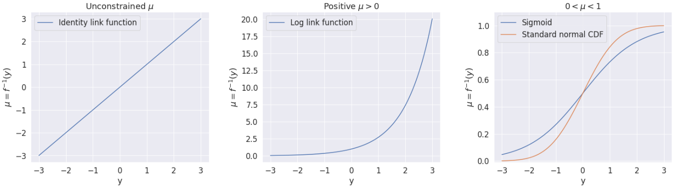
1. We replace the linear model with a Gaussian process (or a NN)

$$f(\boldsymbol{x}) \sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}'))$$

2. The *link function* f that related the mean of the linear model to the mean of the response variable $t(\boldsymbol{x})$

$$g\left[\mu(\boldsymbol{x})\right] = f(\boldsymbol{x}) \qquad \Longleftrightarrow \qquad \mathbb{E}\left[y|\boldsymbol{x}\right] = \mu(\boldsymbol{x}) = g^{-1}\left[f(\boldsymbol{x})\right]$$

3. The distribution $p(y|\boldsymbol{x})$ for the response variable $y(\boldsymbol{x})$, e.g. Poisson, binomial, gamma etc.



Adapting GP models to different likelihoods

- From last weeks' exercise: Laplace approximation GP for classification

$$p(\mathbf{f}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{p(\mathbf{y})} \approx q(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{S}),$$

- The log joint of the target **y** and latent function values **f**

$$\log p(\mathbf{y}, \mathbf{f}) = \log p(\mathbf{y}|\mathbf{f}) + \log p(\mathbf{f}) = \sum_{n=1}^{N} \log p(y_n|f_n) - \frac{N}{2}\log(2\pi) - \frac{1}{2}|\mathbf{K}| - \frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f}$$

- The gradient and Hessian of the log joint

$$\nabla_{\mathbf{f}} \log p(\mathbf{y}, \mathbf{f}) = \sum_{n=1}^{N} \nabla_{\mathbf{f}} \log p(y_n|f_n) - \mathbf{K}^{-1}\mathbf{f},$$

$$\nabla^2_{\mathbf{f}} \log p(\mathbf{y}, \mathbf{f}) = \sum_{n=1}^{N} \nabla^2_{\mathbf{f}} \log p(y_n|f_n) - \mathbf{K}^{-1}$$

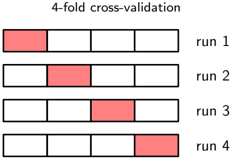
- Hence, all we need to change is the first and second order derivative of log likelihood

Generalization and evaluation

Generalization error and model evaluation

- Consider a supervised problem with $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$

- We measure performance of models using *cross-validation*
  1. Held-out test set
  2. K-fold
  3. Leave-one-out
  4. Split-half
  5. ...



- Goal: assess the *generalization error* of the model, i.e. how well can we expected the model to perform on a new, unseen test example?

- We use cross-validation to *estimate* the generalization error

- Parameter tuning: training/validation/test or nested cross-validation

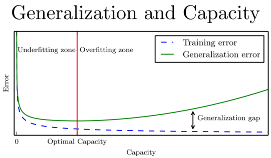


Figure from Goodfellow 2016

Generalization error and loss functions

- Let y be the target value for a input vector $\boldsymbol{x}$ and let $\hat{y} \equiv \hat{y}(\boldsymbol{x})$ be a prediction

- The *loss function* $\mathcal{L}(y, \hat{y})$ define the cost of predicting $\hat{y}$ when the true value is y

  1. The *quadratic loss* for regression is

  $$\mathcal{L}(y, \hat{y}) = (y - \hat{y})^2$$

  2. The *0/1 loss* for classification is given by

  $$\mathcal{L}(y, \hat{y}) = \mathbb{I}[y \neq \hat{y}]$$

  3. ...

- The *generalization error* (or *expected loss*, *risk*, *out-of-sample error*) for a model $\hat{y}(\boldsymbol{x})$ is defined as

$$\mathcal{R}_{\hat{y}} \equiv \mathbb{E}\left[\mathcal{L}(y, \hat{y}(\boldsymbol{x}))\right] = \iint \mathcal{L}(y, \hat{y}(\boldsymbol{x}))p(\boldsymbol{x}, y)\mathrm{d}\boldsymbol{x}\mathrm{d}y$$

- We rarely know the *true data generating mechanism* $p(\boldsymbol{x}, y)$ in practice

Estimating the generalization

- The *exact* generalization error

$$\mathcal{R}_{\hat{y}} \equiv \mathbb{E}\left[\mathcal{L}(y, \hat{y}(\boldsymbol{x}))\right] = \iint \mathcal{L}(y, \hat{y}(\boldsymbol{x}))p(\boldsymbol{x}, y)\mathrm{d}\boldsymbol{x}\mathrm{d}y$$

- For *independent and identically distributed (i.i.d)* samples $(\boldsymbol{x}_i^*, y_i^*) \sim p(\boldsymbol{x}, y)$ for $i = 1, \dots, N_{\text{test}}$

$$\mathcal{R}_{\hat{y}} \approx \widehat{\mathcal{R}}_{\hat{y}}^{\text{test}} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathcal{L}(y_i^*, \hat{y}(\boldsymbol{x}_i^*))$$

- The *estimator* is accurate when the test set is large: $\widehat{\mathcal{R}}_{\hat{y}}^{\text{test}} \to \mathcal{R}_{\hat{y}} \quad \text{for} \quad N_{\text{test}} \to \infty$

- The *empirical risk* for the training set

$$\mathcal{R}_{\hat{y}} \approx \widehat{\mathcal{R}}_{\hat{y}}^{\text{train}} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \mathcal{L}(y_i, \hat{y}(\boldsymbol{x}_i))$$

- Many learning algorithms can be expressed as *empirical risk minimization* (ERM)

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \widehat{\mathcal{R}}_{\hat{y}}^{\text{train}}$$

- For ERM, one can show that the training error is *optimistic*

$$\mathbb{E}\left[\widehat{\mathcal{R}}_{\hat{y}}^{\text{train}}\right] \leq \mathbb{E}\left[\widehat{\mathcal{R}}_{\hat{y}}^{\text{test}}\right]$$

Example: the expected generalization error wrt. to the squared loss

- Consider the following (very simple) linear regression model (no slope, only intercept)

$$y = w_{\text{true}} + e, \qquad e \sim \mathcal{N}(0, \sigma^2)$$

- We will use some estimator $\hat{w}$ for w_{true} (fixed) as a prediction for a new y, i.e. $y^* = \hat{w}$.

- What is the *generalization error* wrt. *squared loss*?

$$\begin{aligned}
\mathcal{R}_{\hat{w}} = \mathbb{E}\left[\mathcal{L}\right] &= \iint (t - \hat{w})^2 p(y, \boldsymbol{x}) \mathrm{d}y \mathrm{d}\boldsymbol{x} \\
&= \int (y - \hat{w})^2 p(y) \mathrm{d}y \\
&= \int (y - \hat{w})^2 \mathcal{N}(y | w_{\text{true}}, \sigma^2) \mathrm{d}y \\
&= \int (y^2 + \hat{w}^2 - 2y\hat{w}) \mathcal{N}(y | w_{\text{true}}, \sigma^2) \mathrm{d}y \\
&= \int y^2 \mathcal{N}(y | w_{\text{true}}, \sigma^2) \mathrm{d}y + \hat{w}^2 \int \mathcal{N}(y | w_{\text{true}}, \sigma^2) \mathrm{d}y - 2\hat{w} \int y \mathcal{N}(y | w_{\text{true}}, \sigma^2) \mathrm{d}y \\
&= (w_{\text{true}} - \hat{w})^2 + \sigma^2
\end{aligned}$$

- Unsurprisingly, the generalization error is minimized when $\hat{w} = w_{\text{true}}$, which leads to $\mathbb{E}\left[\mathcal{L}\right] = \sigma^2$.

What is the expected generalization error for a given dataset size?

- The generalization error

$$\mathcal{R}_{\hat{w}} = (w - \hat{w})^2 + \sigma^2$$

- Suppose we observe a dataset $\mathcal{D} = \{y_n\}_{n=1}^{N}$ with the goal of estimating the parameter w_{true}

$$y_n = w_{\text{true}} + e_n, \qquad e_n \sim \mathcal{N}(0, \sigma^2)$$

- The maximum likelihood (or ERM) estimator for w_{true} and its sampling distribution is

$$w_{\text{MLE}} = \frac{1}{N}\sum_{n=1}^{N} y_n, \qquad w_{\text{MLE}} \sim \mathcal{N}\left(w_{\text{true}}, \frac{1}{N}\sigma^2\right)$$

- The *expected generalization error* for the maximum likelihood estimator for a dataset of size N is

$$\left\langle \mathcal{R}_{w_{\text{MLE}}} \right\rangle_N = \int \left[(w_{\text{true}} - \hat{w}_{\text{ML}})^2 + \sigma^2\right] \mathcal{N}\left(w_{\text{ML}} | w_{\text{true}}, \frac{\sigma^2}{N}\right) \mathrm{d}w_{\text{ML}} = \frac{\sigma^2}{N} + \sigma^2$$

- Interpretation
  1. Average excess error due to finite training data
  2. Inherent noise in the data

Decision theory

Uncertainty in multi-class classification

- Categorical distributions for multi-class classification with K classes

$$y_n|\mathbf{f}_n \sim \text{Categorical}\left[\text{softmax}(\mathbf{f}_n)\right]$$

- The posterior predictive distribution will be another categorial distribution

$$p(y^* = k|\boldsymbol{y}, \boldsymbol{x}^*) = \pi_k \qquad \text{for} \qquad \sum_{k=1}^{K} \pi_k = 1 \qquad \text{and} \qquad 0 \leq \pi_k \leq 1$$

- Predict using most likely class

$$\hat{y}^* = \arg\max_k p(y^* = k|\boldsymbol{y}, \boldsymbol{x}^*)$$

- Example

$$\begin{aligned} p(y^* = \text{cat}|\text{img}) &= 0.36 \\ p(y^* = \text{dog}|\text{img}) &= 0.34 \\ p(y^* = \text{bird}|\text{img}) &= 0.30 \end{aligned} \qquad \begin{aligned} p(y^* = \text{healthy}|\text{x-ray}) &= 0.36 \\ p(y^* = \text{pre-phase cancer}|\text{x-ray}) &= 0.34 \\ p(y^* = \text{severe cancer}|\text{x-ray}) &= 0.30 \end{aligned}$$

Quantifying uncertainty for multi-class classification

- The *confidence* of the posterior predictive distribution is defined as

$$\mathcal{C} = \max_k p(y^* = k | \boldsymbol{y}, \boldsymbol{x}^*) \qquad \text{(range: } \left[\tfrac{1}{K}, 1\right]\text{)}$$

- The *entropy* is defined as (higher ⇒ more uncertainty)

$$\mathcal{H} = -\sum_{k=1}^{K} \pi_k \log \pi_k \qquad \text{(range: } [0, \log K]\text{)}$$

- Convention for entropy calculations: $0 \log 0 = 0$

- Examples

$$\begin{aligned} p(y^* = \text{cat}|\text{img}) &= 0.50 \\ p(y^* = \text{dog}|\text{img}) &= 0.50 \\ p(y^* = \text{bird}|\text{img}) &= 0.0 \end{aligned}$$

$$\begin{aligned} p(y^* = \text{healthy}|\text{x-ray}) &= 0.50 \\ p(y^* = \text{pre-phase cancer}|\text{x-ray}) &= 0.25 \\ p(y^* = \text{severe cancer}|\text{x-ray}) &= 0.25 \end{aligned}$$

$$\mathcal{C} = 0.5 \qquad \mathcal{H} \approx 0.69$$

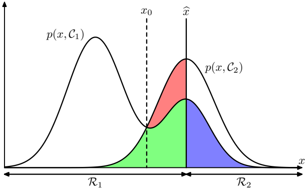
$$\mathcal{C} = 0.5 \qquad \mathcal{H} \approx 1.04$$

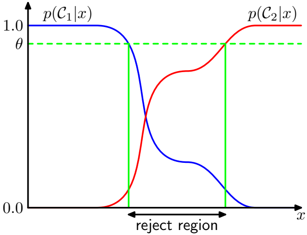
Reject option: I don't know?

- For applications in medicin etc. it may be better to say "I don't know" rather than provide a prediction we don't really trust
- *Reject option*: Avoid making a decision if the uncertainty is too large
- Reject to make decision if

$$\mathcal{C} = \max_k p(y^* = i|\boldsymbol{y}, \boldsymbol{x}^*) < \theta_{\text{reject}}$$

- If $\theta_{\text{reject}} = 1$, then all samples will get rejected
- If $\theta_{\text{reject}} \leq \frac{1}{K}$, then no samples will get rejected





Decision theory: a more formal approach

- In Bayesian modelling we strive to represent all unknown quantities (parameters, predictions etc.) using *probability distributions*

- Yet, we are often forced to reduce these to single decisions/predictions

  1. Cancer or not cancer?
  2. Which online campaign is better? A, B or C etc?
  3. How many airline passengers in year 2025?
  4. Which model best describes the data?
  5. ...

- *Statistical decision theory* tells us to make optimal decisions under uncertainty

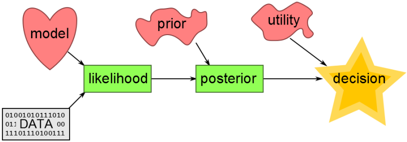


Figure from https://www.azimuthproject.org

Bayesian decision theory in a nutshell

- A decision maker has a set of actions/choices $a \in \mathcal{A}$ to choose from

- The optimal action depends on the true state $s \in \mathcal{S}$ of the system of interest

- The *loss function* $\mathcal{L}(s, a)$ determines the cost for choice a when the true state is s. We can also use a *utility function* $\mathcal{U}(s, a) = -\mathcal{L}(s, a)$

- Example: assigning medical treatment to potential Covid19 patient

$\mathcal{L}(s, a)$	Do nothing	Isolate at home	Hospitalization
Healthy	0	10	50
Covid19, mild symptoms	10	0	20
Covid19, severe symptoms	100	10	0

- In practice, we do not know the true state s. What to do?

- *Bayesian decision theory*

  1. Compute posterior of the state s given data $\boldsymbol{y}$, i.e. $p(s|\boldsymbol{y})$

  2. Choose the action that minimizes the posterior extected loss

$$\hat{a} = \arg\min_{a \in \mathcal{A}} \mathbb{E}_{p(s|\boldsymbol{y})} \left[\mathcal{L}(s, a)\right]$$

5 minutes exercise

$\mathcal{L}(s, a)$	Do nothing	Isolate at home	Hospitalization
Healthy	0	10	50
Covid19, mild symptoms	10	0	20
Covid19, severe symptoms	100	10	0

- Suppose a medical doctor collected data about a patient and arrived at the following posterior predictions for whether the patient will get Covid19 or not

State s	Healthy	Mild symptoms	Severe symptoms
$p(s\|\boldsymbol{y})$	0.65	0.3	0.05

Questions

- What is the posterior expected loss for each action, i.e. $\mathbb{E}_{p(s|\boldsymbol{y})}\left[\mathcal{L}(s, a)\right]$?

- What is the optimal action? ($\hat{a} = \arg\min_{a \in \mathcal{A}} \mathbb{E}_{p(s|\boldsymbol{y})}\left[\mathcal{L}(s, a)\right]$)

Decision theory for classification I

- Consider a binary classification problem with $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$ for $y_i \in \{0, 1\}$
- The *0/1 utility function* is given by $\mathcal{U}(y, \hat{y}(\boldsymbol{x})) = \mathbb{I}\left[y = \hat{y}(\boldsymbol{x})\right]$

$\mathcal{U}(y, \hat{y}(\boldsymbol{x}))$	$\hat{y} = 0$	$\hat{y} = 1$
$y = 0$	1	0
$y = 1$	0	1

- If $p \equiv p(y^* = 1|\boldsymbol{y}, \boldsymbol{x}^*)$ denotes the posterior class probability for input point $\boldsymbol{x}^*$, then

$$\mathbb{E}\left[\mathcal{U}(y^*, \hat{y}(\boldsymbol{x}))\right] = \sum_{y^*} p(y^*|\boldsymbol{y}, \boldsymbol{x}^*)\mathcal{U}(y^*, \hat{y}(\boldsymbol{x})) = (1-p)\mathbb{I}\left[0 = \hat{y}(\boldsymbol{x})\right] + p\mathbb{I}\left[1 = \hat{y}(\boldsymbol{x})\right]$$

- If we choose $\hat{y}(\boldsymbol{x}) = 0$, then

$$\mathbb{E}\left[\mathcal{U}(y^*, 0)\right] = (1-p)\mathbb{I}\left[0 = 0\right] + p\mathbb{I}\left[1 = 0\right] = 1 - p$$

- .. and if we choose $\hat{y}(\boldsymbol{x}) = 1$, then

$$\mathbb{E}\left[\mathcal{U}(y^*, 1)\right] = (1-p)\mathbb{I}\left[0 = 1\right] + p\mathbb{I}\left[1 = 1\right] = p$$

- Picking the class with largest posterior pred. probability is *Bayes optimal* under the 0/1-loss function

Decision theory for classification II

- Consider a binary classification problem with $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$ for $y_i \in \{0, 1\}$
- Example: classifying cancer from medical imagery $\boldsymbol{x}$

$\mathcal{U}(y, \hat{y}(\boldsymbol{x}))$	$\hat{y} = 0$(not cancer)	$\hat{y} = 1$(cancer)
$y = 0$(not cancer)	1	-10
$y = 1$(cancer)	-100	1

- How certain do we need to be before we "dare" to predict $\hat{y}^* = 0$?

$$\begin{aligned} &\mathbb{E}\left[\mathcal{U}(y^*, 0)\right] \geq \mathbb{E}\left[\mathcal{U}(y^*, 1)\right] \\ &\Rightarrow (1-p)\mathcal{U}_{00} + p\mathcal{U}_{10} \geq (1-p)\mathcal{U}_{01} + p\mathcal{U}_{11} \\ &\Rightarrow p \leq \frac{\mathcal{U}_{01} - \mathcal{U}_{00}}{\mathcal{U}_{10} - \mathcal{U}_{11} + \mathcal{U}_{01} - \mathcal{U}_{00}} \end{aligned}$$

- We have

$$p \leq \frac{\mathcal{U}_{01} - \mathcal{U}_{00}}{\mathcal{U}_{10} - \mathcal{U}_{11} + \mathcal{U}_{01} - \mathcal{U}_{00}} = \frac{-10-1}{-100-1-10-1} \approx 0.099$$

- That is, if $p(y^* = 0|\boldsymbol{x}) = 1 - p > 0.901$, we predict "No cancer"

Decision theory for regression I

- Consider a regression problem with $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$ for $y_i \in \mathbb{R}$

- The most common loss function for regression is the *quadratic loss*

$$\mathcal{L}(y, y(\boldsymbol{x})) = (y - \hat{y}(\boldsymbol{x}))^2$$

- The expected loss for a complete probabilistic description $p(\boldsymbol{x}, y)$ is

$$\mathbb{E}\left[\mathcal{L}(y, y(\boldsymbol{x}))\right] = \iint (y - \hat{y}(\boldsymbol{x}))^2 p(\boldsymbol{x}, y) \mathrm{d}\boldsymbol{x} \mathrm{d}y$$

- What is the *optimal predictor function* $\hat{y}(\boldsymbol{x})$?

$$\hat{y}(\boldsymbol{x}) = \mathbb{E}\left[y|\boldsymbol{x}\right]$$

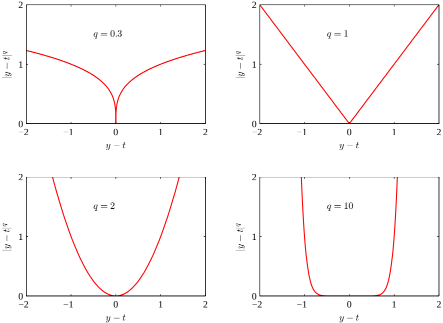
- The *posterior predictive mean* is *optimal* wrt. the quadratic loss

Decision theory for regression II

- The *Minkowski loss* is a generalization of the quadractic loss and is given by $\mathcal{L} = (\hat{y}(\boldsymbol{x}) - y)^q$

- One can show that ...
  - the posterior mean is optimal for $q = 2$
  - the posterior median is optimal for $q = 1$
  - the posterior mode is optimal for $q \to 0$

- $q = 2$ can be sensitive to outliers because of the quadratic form, while $q = 1$ is more robust



Calibration

Calibration for classification models

- We have seen that probabilities play an important role in decision-making, but how do we know if the probabilities are accurate?

- Training models via maximum likelihood or Bayesian methods should in theory yield calibrated models, however, in practice, models are rarely perfectly calibrated

- If we have an independent validation/test set, we can quantify the degree of calibration or miscalibration

- Among all the example in the test set, where the predictive probability is approximately 80% we expect roughly 80% of the corresponding examples to belong to the positive class.

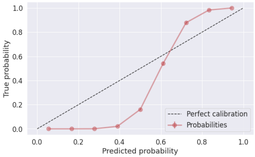
- Metrics for quantifying degree of miscalibration for classification

  1. Expected calibration error (ECE)

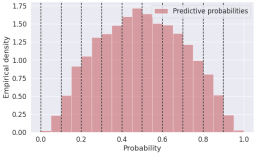
  2. Maximum calibration error

  3. Marginal calibration error

  4. Brier score



Expected calibration error (ECE)



- Compute predictions for validation set $\mathcal{D}_{\text{val}} = \{\mathbf{x}_m^*, y_m^*\}_{m=1}^M$
- Divide unit interval in B bins such that $I_b = \left(\frac{b-1}{B}, \frac{b}{B}\right]$
- Let $\mathcal{B}_b$ be the set of indices of samples whose prediction confidence into interval and define

$$\hat{y}_m^* = \arg\max_c p(y_m^* = c|\mathbf{y}, \mathbf{x}_m^*)$$

$$\hat{p}_m^* = \max_c p(y_m^* = c|\mathbf{y}, \mathbf{x}_m^*)$$

- Then *average accuracy* for bin b is defined as

$$\text{acc}(\mathcal{B}_b) = \frac{1}{|\mathcal{B}_b|} \sum_{m \in \mathcal{B}_b} \mathbb{I}\left[\hat{y}_m^* = y_m^*\right]$$

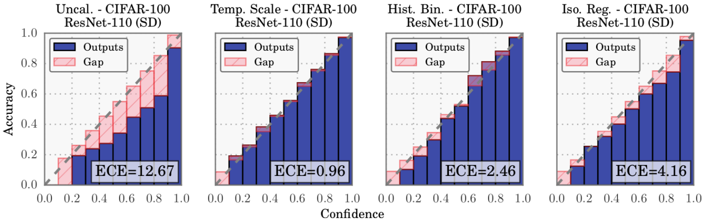
- .. and the *average confidence* for bin b is

$$\text{conf}(\mathcal{B}_b) = \frac{1}{|\mathcal{B}_b|} \sum_{m \in \mathcal{B}_b} \hat{p}_m^*$$

- The *expected calibration error (ECE)* is then given by

$$\text{ECE} = \sum_{b=1}^{B} \frac{|\mathcal{B}_b|}{M} |\text{acc}(\mathcal{B}_b) - \text{conf}(\mathcal{B}_b)|$$

Expected calibration error (ECE)



- Results on CIFAR-100 dataset from Guo et al, 2017 (and page 573 in Murphy2)

- Several methods for recalibration:
  1. Platt scaling
  2. Temperature scaling
  3. Histogram binning
  4. Isotonic regression

- In *temperature scaling*, we introduce a temperature parameter $T > 0$ in the softmax-function:

$$p(y|f(\mathbf{x}), T) = \mathrm{Cat}(y|\mathrm{softmax}\,(f(\mathbf{x})/T))$$

- T is estimated using a validation set.

Summary and take-aways

- Gaussian processes and neural networks are linked in many ways
- A suitable likelihood for your data generally gives better results
- We discussed generalized linear models (GLMs) and extensions to non-linear models like Gaussian process and neural networks
- Our goal is to minimize the *generalization error*, and we *estimate* the generalization error using cross-validation
- We talked about how we can use *confidence* and *entropy* to quantify the uncertainty for multi-class classification
- Bayesian decision theory: Uncertainties are crucial for optimal decision-making and therefore *calibration* becomes important
- The calibration error can be assessed via the *expected calibration error* (ECE) metric or *reliability curves*

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{|\mathcal{B}_b|}{M} |\mathrm{acc}(\mathcal{B}_b) - \mathrm{conf}(\mathcal{B}_b)|$$