Bayesian Methods

Peter Komar

December 11, 2018

Contents

1 Foundations

1.1 Definitions, identities

Notation

- Upper-case letters (A, B, C, X, Y): random variables
- Lower-case letters (a, b, c, x, y): real numbers
- $P(X = x) \;=:\; P(x)$
- $P(X = x \text{ and } Y = y) \;=:\; P(x, y)$
- $P(A = a, \text{ given } B = b) \;=:\; P(a \mid b)$
- $\int_{\infty}^{+\infty} [\ldots] da \;=:\; \sum_{a \in \mathbb{R}} [\ldots] \;=:\; \sum_a [\ldots]$

Conditional probability

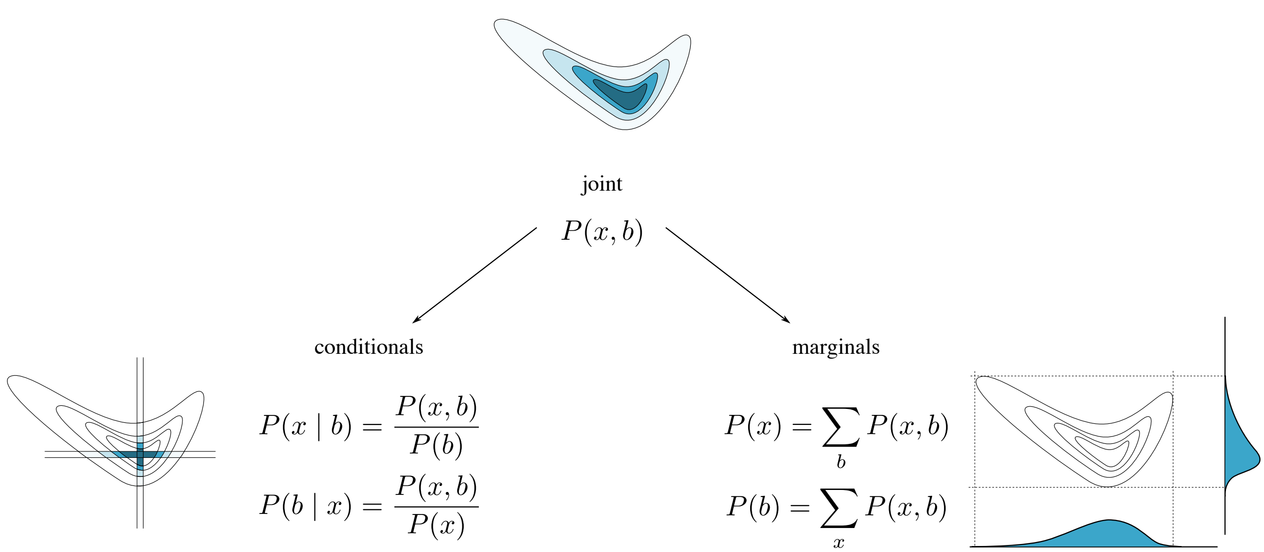
- $P(a \mid b) \;=\; \frac{P(a,b)}{P(b)}$
- $P(a, b) \;=\; P(a \mid b)\, P(b)$
- $P(a, b \mid c) \;=\; P(a \mid b, c)\, P(b \mid c)$
- $\sum_a P(a \mid b) \;=\; 1$, but $\sum_b P(a \mid b) \neq 1$, in general
- $\sum_b P(a \mid b)\, P(b) \;=\; \sum_b P(a, b) \;=\; P(a)$

Marginal

- $P(a) \;=\; \sum_b P(a \mid b)\, P(b)$

Bayes theorem

- $P(b \mid x) \;=\; \frac{1}{P(x)} P(x \mid b)\, P(b)$



1.2 Bayesian inference

Prior, likelihood, posterior

- Data: $D = \{x_1, x_2, \dots x_n\}$, independent measurements.
- Model: M with θ: parameter(s) to estimate
- Prior: $P(\theta)$
- Likelihood: $P(D \mid \theta) = P(x_1 \mid \theta)\; P(x_2 \mid \theta)\; \dots\; P(x_n \mid \theta) = \prod_{i=1}^{n} P(x_i \mid \theta)$
- Unnormalized posterior: $P^*(\theta \mid D) \;=\; P(D \mid \theta)\; P(\theta)$
- Normalization: $Z = \sum_\theta P^*(\theta \mid D)$
- Posterior: $P(\theta \mid D) = \frac{1}{Z} P^*(\theta \mid D)$

Example

"Three light bulbs of the same make lasted 1, 2 and 5 months of continuous use. Let us estimate the lifetime of this kind of light bulb."

- $D = \{t_1, t_2, t_3\} = \{1, 2, 5\}$
- M: Light bulbs have average lifetime of T months.
- $P(T) = \frac{1}{1000}$, uniform on [0,1000].
- $P(t \mid T) = \frac{1}{T} \exp\left(-\frac{t}{T}\right)$
- $P(D \mid T) = \prod_i P(t_i \mid T) = \prod_{i=1}^{3} \frac{1}{T} \exp\left(-\frac{t_i}{T}\right) = \frac{1}{T^3} \exp\left(-\frac{1+2+5}{T}\right)$
- $P^*(T \mid D) = \frac{1}{T^3} \exp\left(-\frac{8}{T}\right)$
- Z and $P(T \mid D)$ can be determined numerically:

```python
import numpy as np

T_arr = np.linspace(0.1, 1000, 10_000)
Pstar_arr = 1.0/T_arr**3 * np.exp(-8/T_arr)
Z = np.sum(Pstar_arr)
P_arr = Pstar_arr / Z
```

  yielding $Z = 0.1562$

- $\mathbb{E}(T \mid D) = \sum_T T\; P(T \mid D)$
- $\mathrm{std}(T \mid D) = \sqrt{\sum_T (T - \mathbb{E}(T))^2\; P(T \mid D)}$

```python
T_ev = np.sum(T_arr * P_arr)
T_std = np.sqrt(np.sum((T_arr - T_ev)**2 * P_arr))
```

  yielding $\mathbb{E}(T \mid D) = 7.937, \quad \mathrm{std}(T \mid D) = 14.48.$

1.3 Model comparison

New definition: Evidence

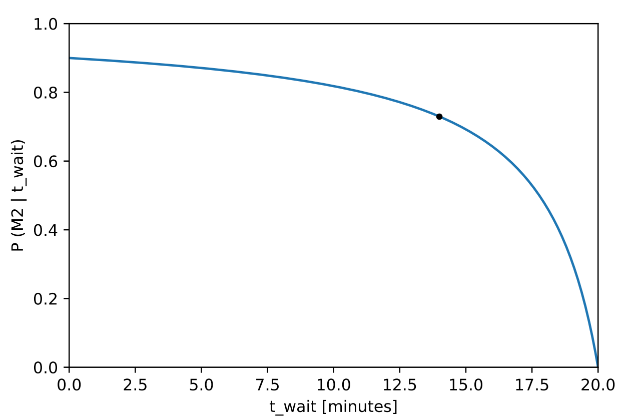
- Data: D
- Model 1: M_1 with parameter θ_1 and prior $P(\theta_1 \mid M_1)$, and likelihood $P(D \mid \theta_1, M_1)$
- Model 2: M_2 with parameter θ_2 and prior $P(\theta_2 \mid M_2)$, and likelihood $P(D \mid \theta_2, M_2)$
- Prior on models: $P(M_1) = 0.5, \quad P(M_2) = 0.5$.
- Evidence for each model: $P(D \mid M_i) = \sum_\theta P(D \mid \theta_i, M_i) P(\theta_i \mid M_i)$
- Unnormalized posterior: $P^*(M_1 \mid D) = P(D \mid M_1)\, P(M_1)$, and $P^*(M_2 \mid D) = P(D \mid M_2)\, P(M_2)$
- Normalization: $Z = P^*(M_1 \mid D) + P^*(M_2 \mid D)$.

Example

"Waiting for my baggage at the airport carousel, there are two possibilities: 1) It could miss the plane, and will never come, or 2) It was on the plane and it had a 1/20 chance of arriving within any of the 1-minute intervals between 0 and 20 minutes. Now, given what is the posterior probability of model 2 given that 14 minutes has passed and the bag has not arrived?"

- $D = \{\text{Bag has not arrived after } t_\text{wait} = 14 \text{ minutes}\}$
- M_1: It never arrives, $P(D \mid M_1) = 1$
- M_2: It shows up some time between 0 and 20 minutes, $P(t_\text{bag} \mid M_2) = 1/20$ for $t_\text{bag} \in [0, 20]$, and the likelihood is $P(D \mid t_\text{bag}, M_2) = 1$, if $t_\text{bag} > t_\text{wait}$, and 0 otherwise.
- $P(M_1) = 0.1, \quad P(M_2) = 0.9$
- $P(D \mid M_1) = 1$
- $P(D \mid M_2) = \sum_{t_\text{bag}} P(D \mid t_\text{bag}, M_2)\; P(t_\text{bag} \mid M_2) = \sum_{t_\text{bag}} [t_\text{bag} > 14] \times \frac{1}{20} = \frac{20-14}{20}$
- $P^*(M_1 \mid D) = 1 \times 0.1, \quad P^*(M_2 \mid D) = \frac{20-14}{20} \times 0.9$
- $Z = 0.1 + \frac{3}{10} \times 0.9 = 0.37$
- $P(M_2 \mid D) = P^*(M_2 \mid D)/Z = 0.7297$.

We can also plot $P(M_2 \mid t_\text{wait})$ for all waiting times between 0 and 20 minutes.



1.4 Prediction

New definition: Predictive distribution

- Data: $D = \{x_1, x_2, \ldots x_n\}$
- Model: M with parameter θ, prior $P(\theta)$ and likelihood $P(x \mid \theta)$
- Posterior: $P(\theta \mid D) = P^*(\theta \mid D)/Z = \ldots$ (see previous sections)
- Predictive distribution: $P(X_{n+1} = x \mid D) = \sum_\theta P(x \mid\theta)\ P(\theta \mid D)$
- Customized prediction: $P(f(\theta) \mid D) = \sum_\theta f(\theta)\ P(\theta \mid D)$

Example

"Two player, A and B are playing a game of luck, where at the beginning of the game a ball is rolled on a pool table to divide the table in two un-equal halves: A's side and B's side. In each subsequent round, a ball is rolled. A point is given to the player on whose side the ball stops. A and B are playing this game until one of them reaches 6 points. The current score is 5 to 3 in favor of A. What is the chance that A will win this game?"

- $D = \{n_A = 5, n_B = 3\}$
- M, first ball: $P(b) = 1$ in $[0, 1]$
- $P(\text{A scores} \mid b) = b$
- $P(D \mid b) = \text{Binomial}(5 \mid 5 + 3,\ b)$
- $P^*(b_0 \mid D) = \text{Binomial}(5 \mid 8, b) \times 1$
- $Z = \sum_b \text{Binomial}(5 \mid 8, b)$ can be calculated numerically

```python
import numpy as np
from scipy.stats import binom

b_arr = np.linspace(0, 1, 1000)
Pstar_arr = binom.pmf(5, 8, b_arr)
Z = np.sum(Pstar_arr)
```

- $P(\text{A wins} \mid b, D) = 1 - P(\text{B wins} \mid b, D) = 1 - (1 - b)^3 = f(b)$
- $P(\text{A wins} \mid D) = \sum_b f(b) P^*(b \mid D)/Z$

```python
P_arr = Pstar_arr / Z
P_Awins = np.sum((1 - (1 - b_arr)**3) * P_arr)
```

yielding $P(\text{A wins} \mid D) = 0.909$

2 Exact inference and Maximum Likelihood Estimate

2.1 Maximum likelihood estimate

MLE-method

- Data: $D = \{x_1, x_2, \dots x_N\}$
- Parameter: θ
- Likelihood: $P(x_i \mid \theta)$
- Total log likelihood: $L(\theta) = \log P(D \mid \theta) = \sum_{i=1}^{N} \log P(x_i \mid \theta)$
- Maximum likelihood estimate $\theta_{\text{MLE}} = \text{argmax}_\theta \log P(D \mid \theta)$,
  Numerically: with gradient descent or EM methods,
  Analytically: equating first derivatives to 0, and solving the system of equations.

Example 1: Normal model

- Data: $D = \{x_i\}_{i=1}^{N}$
- Parameters: $\mu \in \mathbb{R}, \quad \sigma^2 > 0$
- Likelihood: $P(x_i \mid \mu, \sigma^2) = \text{Normal}(x_i \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x_i-\mu)^2}{2\sigma^2}\right]$
- Total log likelihood:

$$L(\mu, \sigma^2) \quad = \quad \sum_{i=1}^{N} \log \text{Normal}(x_i \mid \mu, \sigma^2) = -\frac{N}{2}\log(\sigma^2) - \sum_{i=1}^{N} \frac{(x_i - \mu)^2}{2\sigma^2} + \text{const.}$$

- Analytical solution:

$$\begin{aligned}
0 &= \left[\frac{\partial L}{\partial \mu}\right]_{\text{MLE}} = \left[\sum_{i=1}^{N} \frac{\mu - x_i}{\sigma^2}\right]_{\text{MLE}} && \Rightarrow && \mu_{\text{MLE}} = \frac{1}{N}\sum_{i=1}^{N} x_i. \\
0 &= \left[\frac{\partial L}{\partial (\sigma^2)}\right]_{\text{MLE}} = \left[-\frac{N}{2\sigma^2} + \sum_{i=1}^{N} \frac{(x_i - \mu)^2}{2(\sigma^2)^2}\right]_{\text{MLE}} && \Rightarrow && (\sigma^2)_{\text{MLE}} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_{\text{MLE}})^2
\end{aligned}$$

Example 2: Cauchy distribution

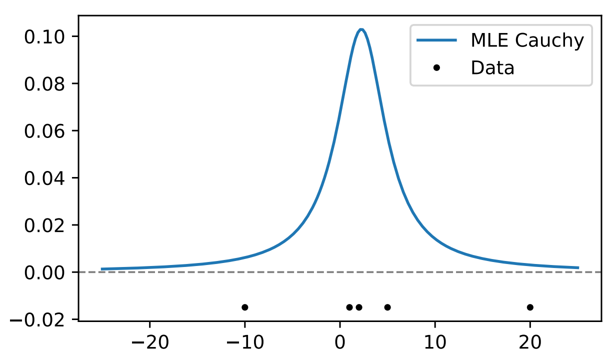
- Data: $D = \{-10, 1, 2, 5, 20\}$
- Parameters: $m \in \mathbb{R}, \quad s > 0.$
- Likelihood: $P(x_i \mid m, s) = \text{Cauchy}(x_i \mid m, s) = \frac{1}{s\pi} \frac{1}{1+[(x_i-m)/s]^2}$
- Total log likelihood:

$$L(m, s) = \sum_{i=1}^{N} \log \text{Cauchy}(x_i \mid m, s) = -N \log(s) - \sum_{i=1}^{N} \log\left(1 + \left[\frac{x_i - m}{s}\right]^2\right)$$

- Numerical maximization (starting from $m_0 = 0$, $s_0 = 10$):

```
import numpy as np
from scipy.optimize import minimize

def cauchy_total_log_likelihood(X, m, s):
    X = np.array(X)

    L = 0
    L += -len(X)/2 * np.log(s**2)
    L += -np.sum( np.log(1 + (X - m)**2 / s**2 ) )

    return L

X = [-10, 1, 2, 5, 20]
def func_to_minimize(theta):
    m = theta[0]
    s = theta[1]
    return - cauchy_total_log_likelihood(X, m, s)

m0 = 0
s0 = 10
result = minimize(func_to_minimize, [m0, s0])
m_MLE, s_MLE = result.x
```

yielding $m_{\text{MLE}} = 2.251$, $s_{\text{MLE}} = 3.090$, the resulting MLE fit is shown below.



2.2 Exact inference examples

Binomial model

- Data: $D = \{(k_1, n_1), (k_2, n_2), \ldots, (k_N, n_N)\}$, where k_i (successes), n_i (attempts) $\in \mathbb{N}$ and $k_i \leq n_i$
- Parameter: p (probability of success) $\in [0, 1]$, flat prior: $P(p) = 1$, on $[0, 1]$
- Likelihood: $P(k_i \mid n_i, p) = \text{Binomial}(k_i \mid n_i, p) = \binom{n_i}{k_i}\, p^{k_i}(1-p)^{n_i - k_i}$
- Posterior:

$$\begin{aligned} P(p \mid D) &= \frac{1}{Z}\prod_{i=1}^{N}\left[p^{k_i}(1-p)^{n_i-k_i}\right] = \frac{1}{Z}p^{k_\text{tot}}(1-p)^{n_\text{tot}-k_\text{tot}} \\ &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}p^{\alpha-1}(1-p)^{\beta-1} = \text{Beta}(p \mid \alpha = k_\text{tot}+1, \beta = n_\text{tot} - k_\text{tot} + 1), \end{aligned}$$

where $k_\text{tot} = \sum_i k_i$ and $n_\text{tot} = \sum_i n_i$. Mean, mode and standard deviation are

$$\begin{aligned} \mathbb{E}(p) = \frac{\alpha}{\alpha+\beta} = \frac{k_\text{tot}+1}{n_\text{tot}+2}, \quad \text{mode}(p) = \frac{\alpha-1}{\alpha+\beta-2} = \frac{k_\text{tot}}{n_\text{tot}}, \\ \text{std}(p) = \frac{\sqrt{\alpha\beta}}{(\alpha+\beta)\sqrt{\alpha+\beta+1}} = \frac{\sqrt{(k_\text{tot}+1)(n_\text{tot}-k_\text{tot}+1)}}{(n_\text{tot}+2)\sqrt{n_\text{tot}+3}} \end{aligned}$$

Poisson model

- Data: $D = \{k_1, k_2, \ldots k_N\}$, where k_i (number of events) $\in \mathbb{N}$
- Parameters: λ (expected number of events) > 0, flat prior: $P(\lambda) = \text{const.}$
- Likelihood: $P(k_i \mid \lambda) = \text{Poisson}(k \mid \lambda) = e^{-\lambda}\frac{\lambda^{k_i}}{k_i!}$
- Posterior:

$$P(\lambda \mid D) = \frac{1}{Z}\prod_{i=1}^{N}\left[e^{-\lambda}\lambda^{k_i}\right] = \frac{1}{Z}e^{-N\lambda}\lambda^{k_\text{tot}} = \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda} = \text{Gamma}(\lambda \mid \alpha = k_\text{tot}+1, \beta = N),$$

where $k_\text{tot} = \sum_i k_i$. Mean, mode and standard deviation are

$$\mathbb{E}(\lambda) = \frac{\alpha}{\beta} = \frac{k_\text{tot}+1}{N}, \quad \text{mode}(\lambda) = \frac{\alpha-1}{\beta} = \frac{k_\text{tot}}{N}, \quad \text{std}(\lambda) = \frac{\sqrt{\alpha}}{\beta} = \frac{\sqrt{k_\text{tot}+1}}{N}$$

Multinomial model

- Data: $D = \{(k_{1,1}, k_{1,2}, \dots k_{1,M}), (k_{2,1}, k_{2,2}, \dots k_{2,M}), \dots (k_{N,1}, k_{N,2}, \dots k_{N,M})\}$, where $k_{i,j}$ (counts of outcome j) $\in \mathbb{N}$, and $\sum_j k_{i,j} = 1$, $\forall i$.
- Parameters: $p = (p_1, p_2, \dots p_M)$, where p_j (probability of outcome j) > 0 and $\sum_j p_j = 1$; flat prior: $P(p) = \text{const.}$
- Likelihood: $P(\{k_{i,j}\}_{j=1}^M \mid p) = \text{Multinomial}(\{k_{i,j}\}_{j=1}^M \mid p) = k_{i,\text{tot}}! \prod_j \frac{p_j^{k_{i,j}}}{k_{i,j}!}$
- Posterior:

$$P(p \mid D) = \frac{1}{Z} \prod_{i=1}^N \prod_{j=1}^M (p_j)^{k_{i,j}} = \frac{1}{Z} \prod_{j=1}^M (p_j)^{k_{\text{tot},j}} = \Gamma(\alpha_\text{tot}) \prod_{j=1}^M \frac{(p_j)^{\alpha_j - 1}}{\Gamma(\alpha_j)} = \text{Dirichlet}(p \mid \alpha_j = k_{\text{tot},j} + 1),$$

where $k_{i,\text{tot}} = \sum_j k_{i,j}$, $k_{\text{tot},j} = \sum_i k_{i,j}$, and $\alpha_\text{tot} = \sum_j \alpha_j = k_{\text{tot},\text{tot}} + M$. Mean, mode and marginal standard deviation are

$$\mathbb{E}(p_j) = \frac{\alpha_j}{\alpha_\text{tot}} = \frac{k_{\text{tot},j} + 1}{k_{\text{tot},\text{tot}} + M}, \quad \text{mode}(p) : \ p_j = \frac{\alpha_j - 1}{\alpha_\text{tot} - M} = \frac{k_{\text{tot},j}}{k_{\text{tot},\text{tot}}}$$

$$\text{std}(p_j) = \frac{\sqrt{\alpha_j(\alpha_\text{tot} - \alpha_j)}}{\alpha_\text{tot}\sqrt{\alpha_\text{tot} + 1}} = \frac{\sqrt{(k_{\text{tot},j} + 1)(k_{\text{tot},\text{tot}} - k_{\text{tot},j} + M - 1)}}{(k_{\text{tot},\text{tot}} + M)\sqrt{k_{\text{tot},\text{tot}} + M + 1}}$$

Exponential model

- Data: $D = \{t_1, t_2, \dots t_N\}$, where t_i (waiting times) > 0
- Parameter: γ (rate) > 0, flat prior: $P(\gamma) = \text{const.}$
- Likelihood: $P(t_i \mid \gamma) = \text{Exponential}(t_i \mid \gamma) = \gamma e^{-\gamma t_i}$
- Posterior:

$$P(\gamma \mid D) = \frac{1}{Z} \prod_{i=1}^N \left[\gamma e^{-\gamma t_i}\right] = \frac{1}{Z} \gamma^N e^{-\gamma t_\text{tot}} = \frac{\beta^\alpha}{\Gamma(\alpha)} \gamma^{\alpha - 1} e^{-\beta\gamma} = \text{Gamma}(\gamma \mid \alpha = N + 1, \beta = t_\text{tot}),$$

where $t_\text{tot} = \sum_i t_i$. Mean, mode, standard deviation are

$$\mathbb{E}(\gamma) = \frac{\alpha}{\beta} = \frac{N + 1}{t_\text{tot}}, \quad \text{mode}(\gamma) = \frac{\alpha - 1}{\beta} = \frac{N}{t_\text{tot}}, \quad \text{std}(\gamma) = \frac{\sqrt{\alpha}}{\beta} = \frac{\sqrt{N + 1}}{t_\text{tot}}$$

Normal

- Data: $D = \{x_1, x_2, \dots x_N\}$, where x (value) $\in \mathbb{R}$
- Parameters: μ (expected value) $\in \mathbb{R}$, σ^2 (variance) > 0, uninformative prior: $P(\mu, \sigma^2) = \text{const.}$
- Likelihood: $P(x_i \mid \mu, \sigma^2) = \text{Normal}(x_i \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x_i-\mu)^2}{2\sigma^2}\right]$
- Posterior:

$$
\begin{aligned}
P(\mu, \sigma^2 \mid D) &= \frac{1}{Z} \prod_{i=1}^{N} \left\{ \frac{1}{\sqrt{\sigma^2}} \exp\left[-\frac{(x_i-\mu)^2}{2\sigma^2}\right] \right\} = \frac{1}{Z} \left(\frac{1}{\sigma^2}\right)^{N/2} \exp\left[-\frac{Ns^2 + N(\mu-m)^2}{2\sigma^2}\right] \\
&= \frac{\sqrt{\lambda}}{\sqrt{2\pi\sigma^2}} \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{1}{\sigma^2}\right)^{\alpha+1} \exp\left[-\frac{2\beta + \lambda(\mu-\mu_c)^2}{2\sigma^2}\right] \\
&= \text{Normal-Inverse-Gamma}\Big(\mu, \sigma^2 \,\Big|\, \alpha = \frac{N-3}{2},\ \beta = \frac{Ns^2}{2},\ \mu_c = m,\ \lambda = N\Big),
\end{aligned}
$$

where $m = \frac{1}{N}\sum_i x_i$ is the empirical mean, $s^2 = \frac{1}{N}\sum_i (x_i - m)^2$ is the empirical variance. The mode is identical to the MLE result

$$\text{mode}(\mu, \sigma^2) = (m, s^2).$$

The marginal, and mean, mode and standard deviation of μ is

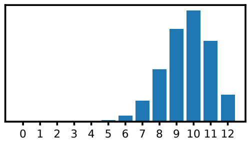
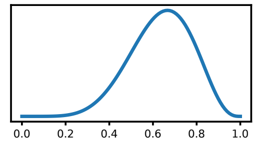
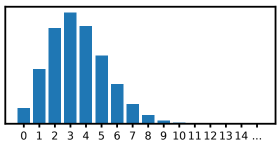
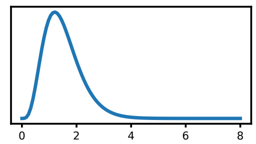
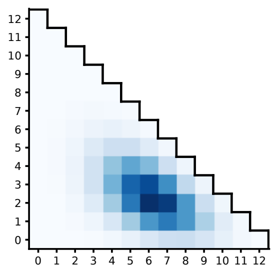
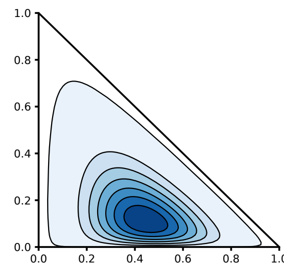
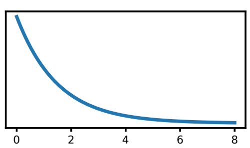
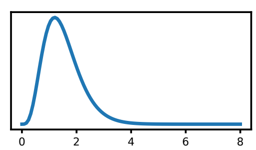
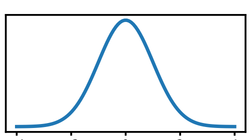
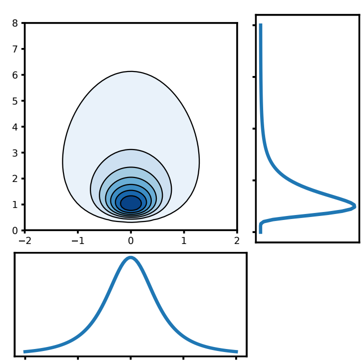
$$
\begin{aligned}
P(\mu \mid D) &= \sum_{\sigma^2} P(\mu, \sigma^2 \mid D) = \frac{\Gamma\left(\frac{N-2}{2}\right)}{\Gamma\left(\frac{N-3}{2}\right)} \frac{1}{\sqrt{\pi s^2}} \left[1 + \frac{(\mu-m)^2}{s^2}\right]^{-(N-2)/2} \\
&= \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \frac{1}{\sqrt{\pi\nu}} \left[1 + \frac{1}{\nu}\left(\frac{\mu - \text{loc}}{\text{scale}}\right)^2\right]^{-(\nu+1)/2} \frac{1}{\text{scale}} \\
&= \text{t-distr}\Big(\mu \,\Big|\, \text{loc} = m,\ \text{scale} = \frac{s}{\sqrt{N-3}},\ \nu = N-3\Big),
\end{aligned}
$$

$$\mathbb{E}(\mu) = m, \quad \text{mode}(\mu) = m, \quad \text{std}(\mu) = \frac{s}{\sqrt{N-3}}\sqrt{\frac{\nu}{\nu-2}} = \frac{s}{\sqrt{N-5}},$$

where ν is the "degrees of freedom" of the Student's t-distribution. The marginal, and mean, mode and standard deviation of σ^2 is

$$
\begin{aligned}
P(\sigma^2 \mid D) &= \sum_{\mu} P(\mu, \sigma^2 \mid D) = \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{1}{\sigma^2}\right)^{\alpha+1} \exp\left(-\frac{\beta}{\sigma^2}\right) \\
&= \text{Inverse-Gamma}\Big(\sigma^2 \,\Big|\, \alpha = \frac{N-3}{2},\ \beta = \frac{Ns^2}{2}\Big)
\end{aligned}
$$

$$\mathbb{E}(\sigma^2) = \frac{\beta}{\alpha-1} = s^2 \frac{N}{N-5}, \quad \text{mode}(\sigma^2) = \frac{\beta}{\alpha+1} = s^2 \frac{N}{N-1}, \quad \text{std}(\sigma^2) = \frac{\beta}{(\alpha-1)\sqrt{\alpha-2}} = s^2 \frac{\sqrt{2}N}{(N-5)\sqrt{N-7}}$$

Model		Posterior	
Binomial		Beta	
Poisson		Gamma	
Multinomial		Dirichlet	
Exponential		Gamma	
Normal		Normal-Inverse-Gamma	 Inverse-Gamma Student's t

3 Priors, Regularization, AIC, BIC, LRT

3.1 Improper and proper priors

Improper priors are not normalizable, i.e. $\sum_\theta P(\theta) = \infty$

- Flat prior: $P(\theta) =$ const. over any infinite domain.
- Uninformative priors (from transformation invariance or max-entropy principles)
  - Location parameter: $P(m) = \text{const.}$, on $m \in (-\infty, +\infty)$,
  - Scale parameter: $P(s) = \frac{\text{const.}}{s}$, on $s \in (0, \infty)$,
  - Probability parameter: $P(p) = \frac{\text{const.}}{p(1-p)}$, on $p \in (0, 1)$.

Proper priors are normalized, i.e. $\sum_\theta P(\theta) = 1$.

3.2 Regularization

Regularized model training

- Data: D,
- Model: M with parameters θ, likelihood $P(D \mid \theta)$
- Cost function $= -\log P(D \mid \theta) = -L(\theta)$, the log likelihood
- Penalty: penalty(θ), which is high for implausible θ values.
- Regularized optimum: $\theta_{\text{reg.opt.}} = \arg\min_\theta(-L(\theta) + \text{penalty}(\theta))$

Example: Linear regression

- Data: $D = \{\ (\{x_{i,k}\}_{k=1}^K,\ y_i)\ \}_{i=1}^N$, where x_i (feature vector) $\in \mathbb{R}^K$, y_i (predicted variable) $\in \mathbb{R}$.
- Parameters: $\{b_k\}_{k=1}^K$, where b_k (coefficient or weight) $\in \mathbb{R}$.
- Model:

$$\begin{aligned}
y_i &= \sum_{k=1}^K x_{i,k} b_k + \varepsilon_i, \quad \text{with} \quad P(\varepsilon_i) = \text{Normal}(\varepsilon_i \mid \mu = 0, \sigma^2 = \sigma^2) \\
y &= Xb + \varepsilon \\
&\quad \text{or equivalently} \\
P(y \mid X, b) &= \prod_{i=1}^N \text{Normal}\left(y_i \;\middle|\; \mu = (Xb)_i,\ \sigma^2 = \sigma^2\right)
\end{aligned}$$

- Log likelihood: $L(b) = \log P(y \mid X, b) = -\frac{N}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^N \left[y_i - \sum_k x_{i,k} b_k\right]^2$
- Maximum likelihood estimate: $b_{\text{MLE}} = \arg\max_b L(b) = (X^\top X)^{-1} X^\top y, \quad (\sigma^2)_{\text{MLE}} = \frac{1}{N}||y - Xb||^2$
- Regularization:
  - "L1 regularization": penalty$(b) = \alpha_1 \sum_k |b_k|$
    $\Leftrightarrow$ Laplace prior: $P(b_k) = \text{const.} \times e^{-\alpha_1 |b_k|} = \text{Laplace}(b_k \mid \text{loc} = 0, \text{scale} = 1/\alpha_1)$
  - "L2 regularization": penalty$(b) = \frac{\alpha_2}{2}\sum_k (b_k)^2$
    $\Leftrightarrow$ Normal prior: $P(b_k) = \text{const.} \times e^{-\alpha_2 (b_k)^2/2} = \text{Normal}(b_k \mid \mu = 0, \sigma^2 = 1/\alpha_2)$
  - "Elastic net regularization": penalty$(b) = \alpha_1 \sum_k |b_k| + \frac{\alpha_2}{2}\sum_k (b_k)^2$

  The "hyperparameters" α_1 and α_2 can be optimized using "Leave-one-out" or "M-fold"cross-validation.

3.3 Model comparison with asymptotic metrics

Maximum likelihood results from two models

- Data $D = \{x_i\}_{i=1}^N$
- Null model: M_0 with parameters θ_0, and $L_0(\theta_0) = P(D \mid \theta_0, M_0)$, $\theta_{0,\mathrm{MLE}} = \mathrm{argmax}_{\theta_0} L_0(\theta_0)$
- Alternate model: M_1 with parameters θ_1, and $L_1(\theta_1) = P(D \mid \theta_1, M_1)$, $\theta_{1,\mathrm{MLE}} = \mathrm{argmax}_{\theta_1} L_1(\theta_1)$

Akaike Information Criterion (AIC)

- $\mathrm{AIC}(M_i) = -2\Big[L_i(\theta_{i,\mathrm{MLE}}) - \dim(\theta_i)\Big]$ for both $i = 0, 1$ models.
- If $\mathrm{AIC}(M_1) < \mathrm{AIC}(M_0)$, then M_1 is more plausible.

Bayesian Information Criterion (BIC)

- $\mathrm{BIC}(M_i) = -2\left[L_i(\theta_{i,\mathrm{MLE}}) - \frac{\ln(N)}{2}\dim(\theta_i)\right]$ for both $i = 0, 1$ models.
- If $\mathrm{BIC}(M_1) < \mathrm{BIC}(M_0)$, then M_1 is more plausible.

Likelihood Ratio Test (LRT)

- $\mathrm{logLR} = \log \frac{P(D \mid M_1, \theta_{1,\mathrm{MLE}})}{P(D \mid M_0, \theta_{0,\mathrm{MLE}})} = L_1(\theta_{1,\mathrm{MLE}}) - L_0(\theta_{0,\mathrm{MLE}})$
- LRT pvalue $= 1 - \mathrm{cdf}\ \chi^2\Big(2\,\mathrm{logRL}\ \Big|\ \mathrm{dof} = \dim(\theta_1) - \dim(\theta_0)\Big)$, where $\mathrm{cdf}\ \chi^2(\ldots \mid \mathrm{dof} = d)$ is the cumulative distribution function of the χ^2 distribution with degrees of freedom d.

Model evidence

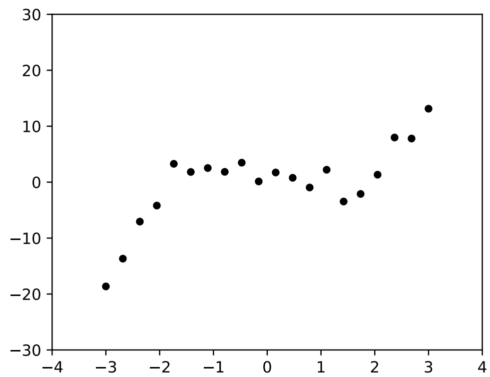
- $P(D \mid M_i) = \sum_{\theta_i} P(D \mid \theta_i, M_i) \approx \exp\left(-\frac{1}{2}\ \mathrm{IC}\right)$,
- where IC can be either AIC or BIC (or WAIC, WBIC)
- Under uniform prior (i.e. $P(M_0) = P(M_1)$), the posterior probability of the alternate model being correct is

$$P(M_1 \mid D) \approx \frac{\exp\left(-\frac{1}{2}\ \mathrm{IC}_1\right)}{\exp\left(-\frac{1}{2}\ \mathrm{IC}_0\right) + \exp\left(-\frac{1}{2}\ \mathrm{IC}_1\right)}$$

Example: Linear regression

- Data: $D = \{(x_i, y_i)\}_{i=1}^N$ (generated from $y = 1 - 3x - x^2/2 + x^3 + \varepsilon$ with $\mathrm{std}(\varepsilon) = 2$).

```
import numpy as np
from numpy.polynomial.polynomial import polyval
from scipy.stats import norm

c_true = [1, -3, -0.5, 1]
sigma_true = 2
x_data = np.linspace(-3, 3, 20)
y_data = [polyval(x, c_true) + norm.rvs(loc=0, scale=sigma_true)
          for x in x_data]
```



- Models: M_K : $K = 0, 1, 2, \ldots$-degree polynomial, parameters: $c = (c_0, c_1, \ldots c_K)$.
- "Linear features": $X_i := (1, x_i, (x_i)^2, (x_i)^3), \ldots (x_i)^K)$.

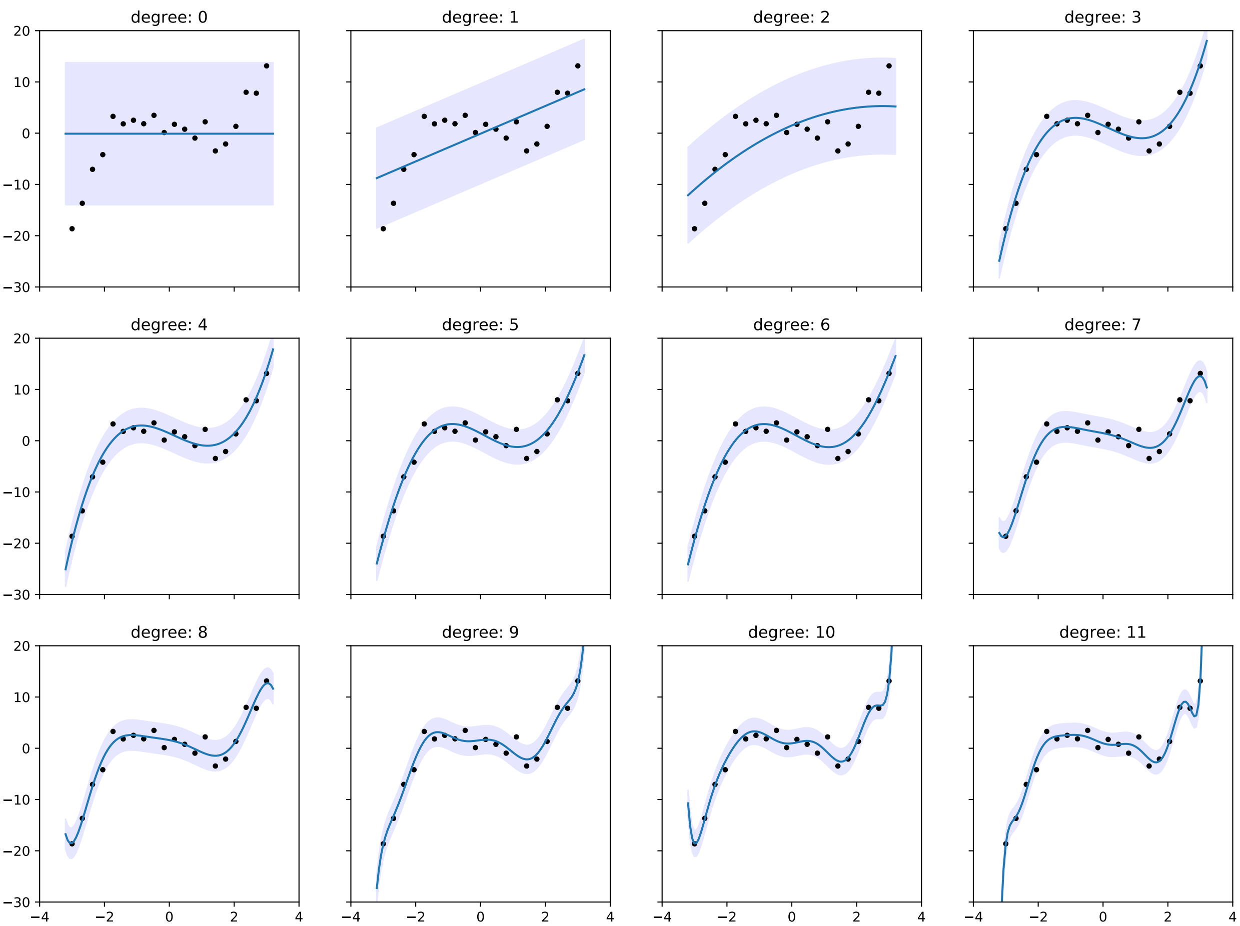
```
def generate_polynomial_features(x_data, degree):
    K = degree
    N = len(x_data)
    X = np.zeros([N, K+1])
    for i, x in enumerate(x_data):
        for k in range(0, K+1, 1):
            X[i,k] = x**k
    return X
```

- Likelihood: $y = \sum_{k=0}^K X_{i,k} c_k + \varepsilon$, with $P(\varepsilon) = \mathrm{Normal}(\varepsilon \mid 0, \sigma^2)$

```
def log_likelihood(X, y, c, sigma2):
    N = len(y_data)
    log_like = 0
    log_like += - N/2.0 * np.log(sigma2)
    log_like += - 1.0/(2 * sigma2) * vector_norm(y - X.dot(c))**2
    return log_like
```

- MLE solution: $c_{\mathrm{MLE}} = (X^\top X)^{-1} X^\top y, \quad (\sigma^2)_{\mathrm{MLE}} = \frac{1}{N} ||y - X c_{\mathrm{MLE}}||^2$

```
def fit_MLE_ploynomial(X, y):
    c_MLE = inv(X.T.dot(X)).dot(X.T).dot(y_data)
    sigma2_MLE = 1.0/len(y) * vector_norm(y - X.dot(c_MLE))**2
    return c_MLE, sigma2_MLE
```



- $\text{AIC}_k = -2[L_k - (k+2)]$

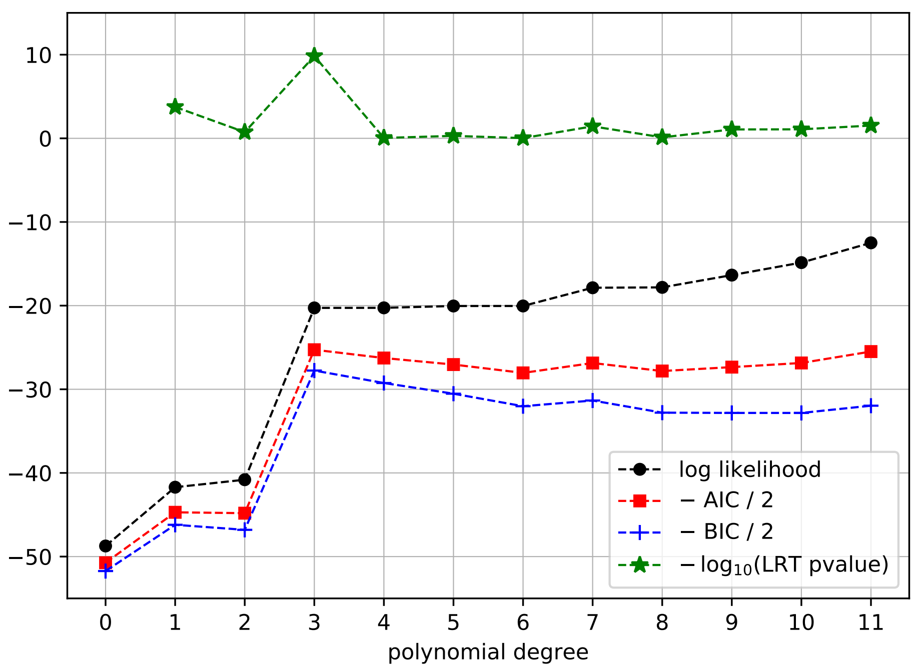
```
def AIC(X, y, c, sigma2):
        dim = len(c) + 1
        loglike = log_likelihood(X, y, c, sigma2)
        return -2 * (loglike - dim)
```

- $\text{BIC}_k = -2[L_k - \frac{\log N}{2}(k+2)]$

```
def BIC(X, y, c, sigma2):
        N = len(y)
        dim = len(c) + 1
        loglike = log_likelihood(X, y, c, sigma2)
        return -2 * (loglike - np.log(N)/2.0 * dim)
```

- LRT $\text{pvalue}_k = 1 - \text{cdf}\ \chi^2(2(L_k - L_{k-1}) \mid \text{dof} = 1)$
  (calculated against the model with one less degree)

```
from scipy.stats import chi2

pvalues = [np.nan]
for deg in degrees[1:]:
    L1 = loglikes[deg]
    L0 = loglikes[deg-1]
    logLR = L1 - L0
    dof = 1
    pvalue = chi2.sf(2*logLR, dof)
    pvalues.append(pvalue)
```



- BIC weights, $P(D \mid M_k) \approx e^{-\mathrm{BIC}_k/2} / \sum_{k'=0}^{K} e^{-\mathrm{BIC}_{k'}/2}$

```
def BIC_weigths(BICs):
        BICs = np.array(BICs)
        w = BICs - np.min(BICs) # for numerical stability
        w = np.exp(-0.5*(w))
        w /= np.sum(w)
        return w
```