Bayesian Machine Learning

Prof. Seungchul Lee
Industrial AI Lab.

Bayesian Decision Theory 1:

Classification

Binary Classification with Gaussian

- Suppose the data $x \in \mathbb{R}$ in 1 D.
- Assume we have two classes (C_1 and C_2) with the probability density functions (pdf) and their cumulative distribution functions (cdf).

$$f_1(x) = \frac{\partial F_1(x)}{\partial x}$$

$$f_2(x) = \frac{\partial F_2(x)}{\partial x}$$

- We further assume two classes are Gaussian distributed and $\mu_1 < \mu_2$.
- Then an instance $x \in \mathbb{R}$ belongs to one of the these two classes:

$$x \sim \begin{cases} \mathcal{N}(\mu_1, \sigma_1^2), & \text{if } x \in \mathcal{C}_1 \\ \mathcal{N}(\mu_2, \sigma_2^2), & \text{if } x \in \mathcal{C}_2 \end{cases}$$

Optimal Boundary for Classes

- Since this is a binary classification problem in 1 dimensional space, we have to determine the threshold ω where $\mu_1 < \omega < \mu_2$. Then

$$\begin{cases} \text{if } x < \omega, & x \in \mathcal{C}_1 \\ \text{if } x > \omega, & x \in \mathcal{C}_2 \end{cases}$$

- We want to minimize a misclassification rate (or error)

$$\begin{aligned} P(\text{error}) &= P(x > \omega, x \in \mathcal{C}_1) + P(x < \omega, x \in \mathcal{C}_2) \\ &= P(x > \omega \mid x \in \mathcal{C}_1)P(x \in \mathcal{C}_1) + P(x < \omega \mid x \in \mathcal{C}_2)P(x \in \mathcal{C}_2) \\ &= (1 - F_1(\omega))\ \pi_1 + F_2(\omega)\ \pi_2 \end{aligned}$$

- where

$$\begin{aligned} P(x \in \mathcal{C}_1) &= \pi_1 \\ P(x \in \mathcal{C}_2) &= \pi_2 \end{aligned}$$

Minimum Error Rate Classification

- Minimize

$$\min_{\omega} P(\text{error}) = \min_{\omega} \{(1 - F_1(\omega))\ \pi_1 + F_2(\omega)\, \pi_2\}$$

- We take derivatives

$$\frac{\partial P(\text{error})}{\partial \omega} = -\ f_1(\omega)\, \pi_1 + f_2(\omega)\, \pi_2 = 0$$

$$\implies\ f_1(\omega)\, \pi_1 = f_2(\omega)\, \pi_2$$

Posterior Probabilities

- Another way is equating the posterior probabilities to have the equation of the classification boundary.
- For x on the boundary

$$P(x \in \mathcal{C}_1 \mid X = x) = P(x \in \mathcal{C}_2 \mid X = x)$$

$$\frac{P(X = x \mid x \in \mathcal{C}_1)P(x \in \mathcal{C}_1)}{P(X = x)} = \frac{P(X = x \mid x \in \mathcal{C}_2)P(x \in \mathcal{C}_2)}{P(X = x)}$$

$$f_1(x)\,\pi_1 = f_2(x)\,\pi_2$$

Boundaries for Gaussian

- Now let us think of data as multivariate Gaussian distributions, $x \sim \mathcal{N}(\mu, \Sigma)$

$$f(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

- Then the equation of boundary

$$\frac{1}{\sqrt{(2\pi)^d |\Sigma_1|}} \exp\left(-\frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1)\right) \pi_1 = \frac{1}{\sqrt{(2\pi)^d |\Sigma_2|}} \exp\left(-\frac{1}{2}(x-\mu_2)^T \Sigma_2^{-1} (x-\mu_2)\right) \pi_2$$

- Two cases
  - Equal covariance
  - Not equal covariance

Equal Covariance

- $\Sigma_1 = \Sigma_2 = \Sigma$

$$\frac{1}{\sqrt{(2\pi)^d|\Sigma|}}\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)\pi_1 = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}}\exp\left(-\frac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)\right)\pi_2$$

$$\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)\pi_1 = \exp\left(-\frac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)\right)\pi_2$$

$$-(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) + 2\ln\pi_1 = -(x-\mu_2)^T\Sigma^{-1}(x-\mu_2) + 2\ln\pi_2$$

$$-x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu_1 + \mu_1\Sigma^{-1}x - \mu_1^T\Sigma^{-1}\mu_1 + 2\ln\pi_1 = -x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu_2 + \mu_2\Sigma^{-1}x - \mu_2^T\Sigma^{-1}\mu_2 + 2\ln\pi_2$$

$$2\left(\Sigma^{-1}(\mu_2-\mu_1)\right)^T x + \left(\mu_1^T\Sigma^{-1}\mu_1 - \mu_2^T\Sigma^{-1}\mu_2\right) + 2\ln\frac{\pi_2}{\pi_1} = a^T x + b = 0$$

- If the covariance matrices are equal, the decision boundary of classification is a line.

Not Equal Covariance

- $\Sigma_1 \neq \Sigma_2$

$$\frac{1}{\sqrt{(2\pi)^d|\Sigma_1|}}\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right)\pi_1 = \frac{1}{\sqrt{(2\pi)^d|\Sigma_2|}}\exp\left(-\frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\right)\pi_2$$

$$\frac{1}{\sqrt{(|\Sigma_1|}}\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right)\pi_1 = \frac{1}{\sqrt{(|\Sigma_2|}}\exp\left(-\frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\right)\pi_2$$

$$-\ln(|\Sigma_1|) - (x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) + 2\ln\pi_1 = -\ln(|\Sigma_2|) - (x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2) + 2\ln\pi_2$$

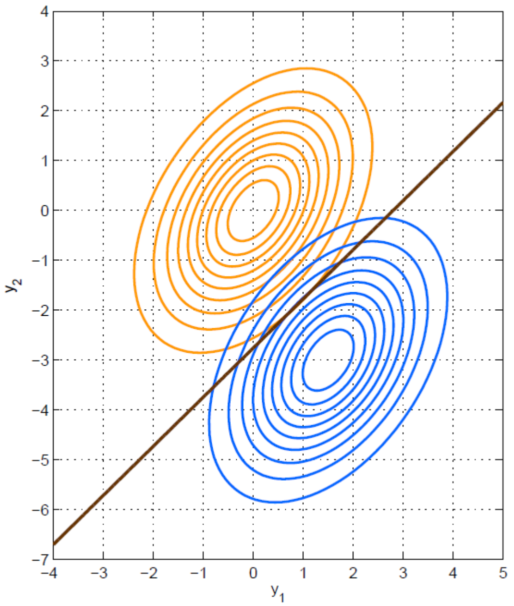
$$-\ln(|\Sigma_1|) - x^T\Sigma_1^{-1}x + x^T\Sigma_1^{-1}\mu_1 + \mu_1\Sigma_1^{-1}x - \mu_1^T\Sigma_1^{-1}\mu_1 + 2\ln\pi_1 = -\ln(|\Sigma_2|) - x^T\Sigma_2^{-1}x + x^T\Sigma_2^{-1}\mu_2 + \mu_2\Sigma_2^{-1}x - \mu_2^T\Sigma_2^{-1}\mu_2 + 2\ln\pi_2$$

$$x^T(\Sigma_1 - \Sigma_2)^{-1}x + 2\left(\Sigma_2^{-1}\mu_2 - \Sigma_1^{-1}\mu_1\right)^T x + \left(\mu_1^T\Sigma_1^{-1}\mu_1 - \mu_2^T\Sigma_2^{-1}\mu_2\right) - \ln\frac{|\Sigma_2|}{|\Sigma_1|} + 2\ln\frac{\pi_2}{\pi_1} = x^TAx + b^Tx + b = 0$$

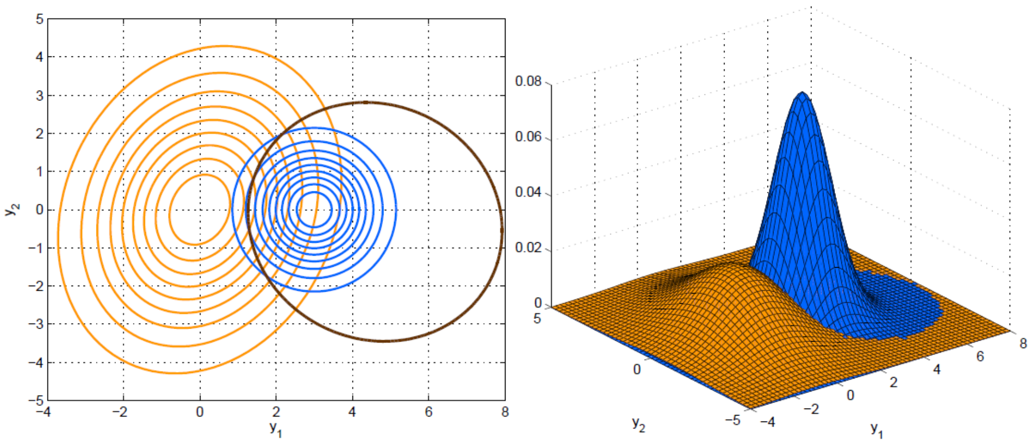
- If the covariance matrices are not equal, the decision boundary of classification is a quadratic.
- When we assume a linear model for any given data set, we should be careful.

Examples of Gaussian Decision Regions

- When the covariances are all equal, the separating surfaces are hyperplanes



- When the covariances are not equal, the separating surfaces are quadratic functions.



Bayesian Decision Theory 2:

Classification

Bayesian Classifier

- Given the height x of a person, decide whether the person is male ($y = 1$) or female ($y = 0$).

- Binary classes: $y \in \{0,1\}$

$$P(y=1 \mid x) = \frac{P(x \mid y=1)P(y=1)}{P(x)} = \frac{\overbrace{P(x \mid y=1)}_{\text{likelihood}}\overbrace{P(y=1)}_{\text{prior}}}{\underbrace{P(x)}_{\text{marginal}}}$$

$$P(y=0 \mid x) = \frac{P(x \mid y=0)P(y=0)}{P(x)}$$

- Decision

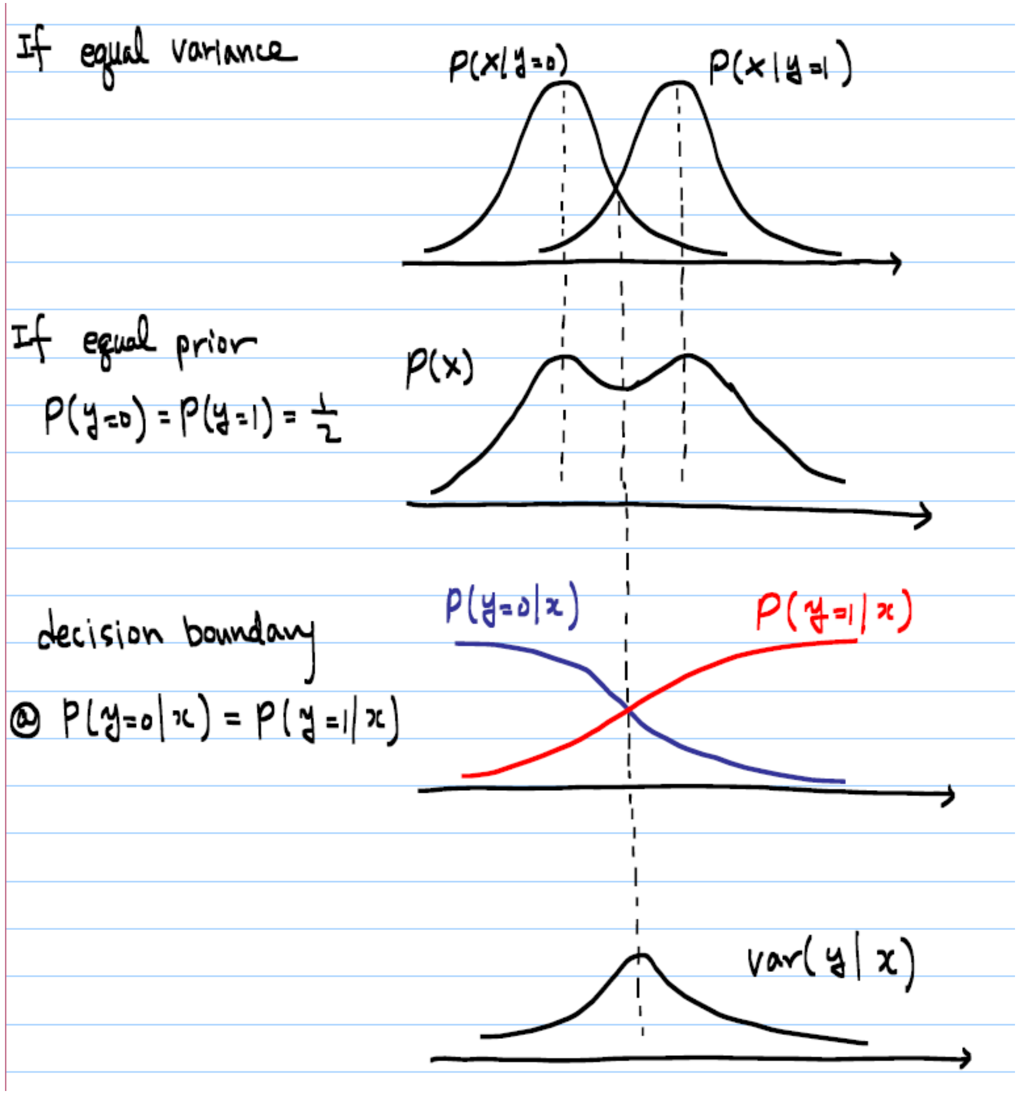
$$\text{If } P(y=1 \mid x) > P(y=0 \mid x), \text{ then } \hat{y} = 1$$
$$\text{If } P(y=1 \mid x) < P(y=0 \mid x), \text{ then } \hat{y} = 0$$

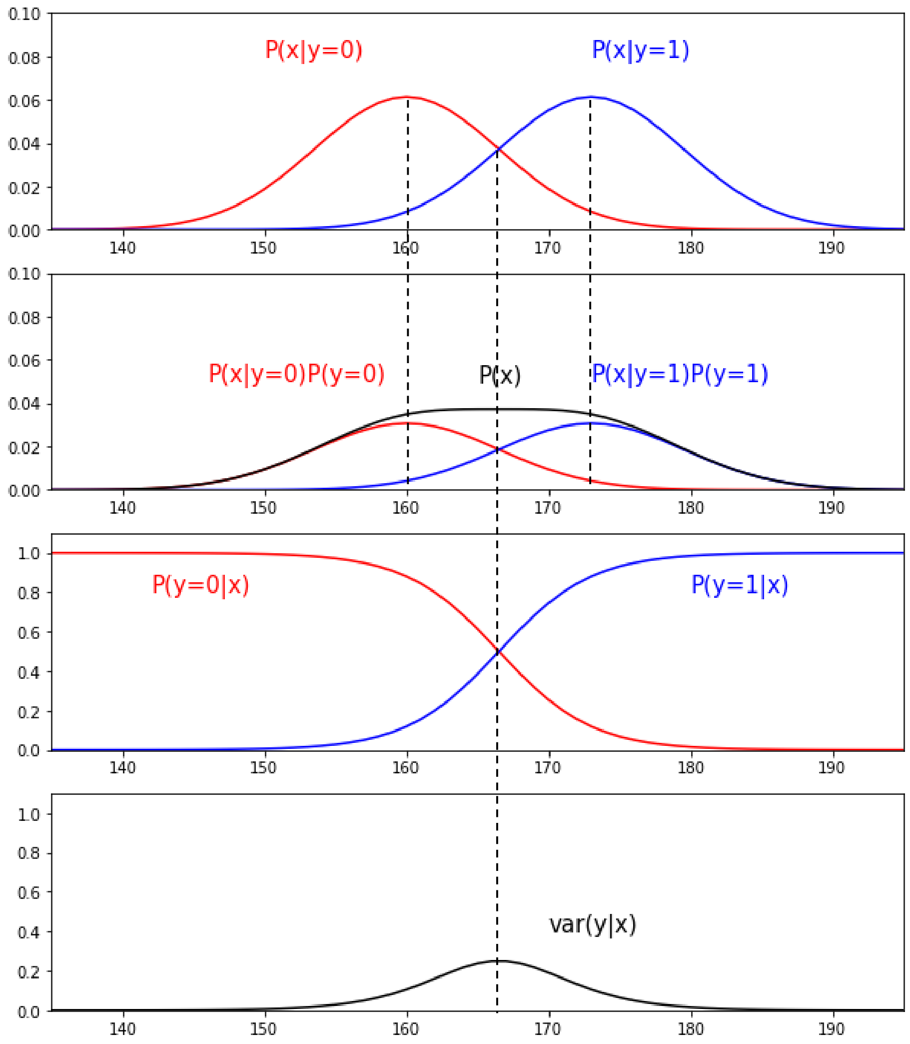
$$\therefore \frac{P(x \mid y=0)P(y=0)}{P(x \mid y=1)P(y=1)} \begin{cases} > 1 & \implies \hat{y} = 0 \\ = 1 & \implies \text{decision boundary} \\ < 1 & \implies \hat{y} = 1 \end{cases}$$

Bayesian Classifier

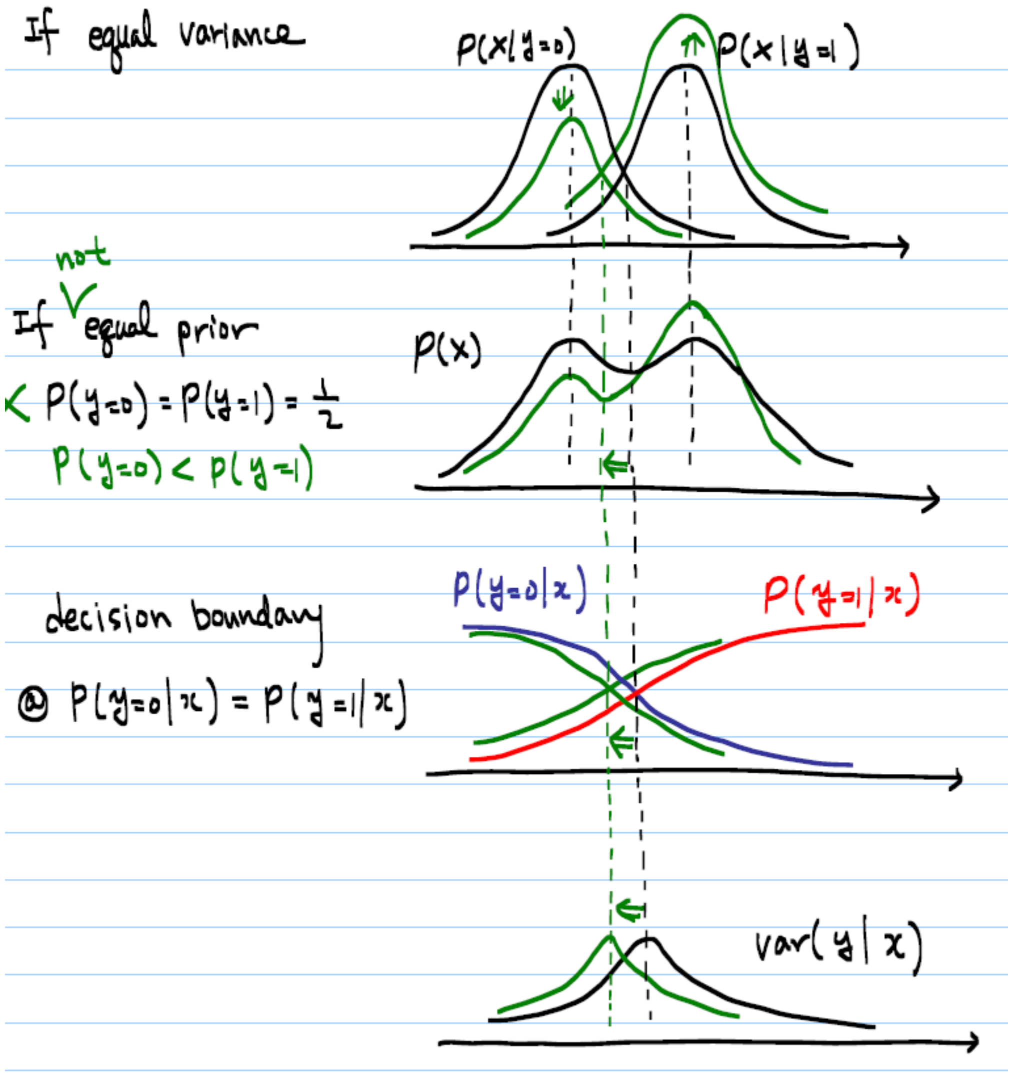
- Equal variance and equal prior
- Equal variance and not equal prior
- Not equal variance and equal prior
- Not equal variance and not equal prior

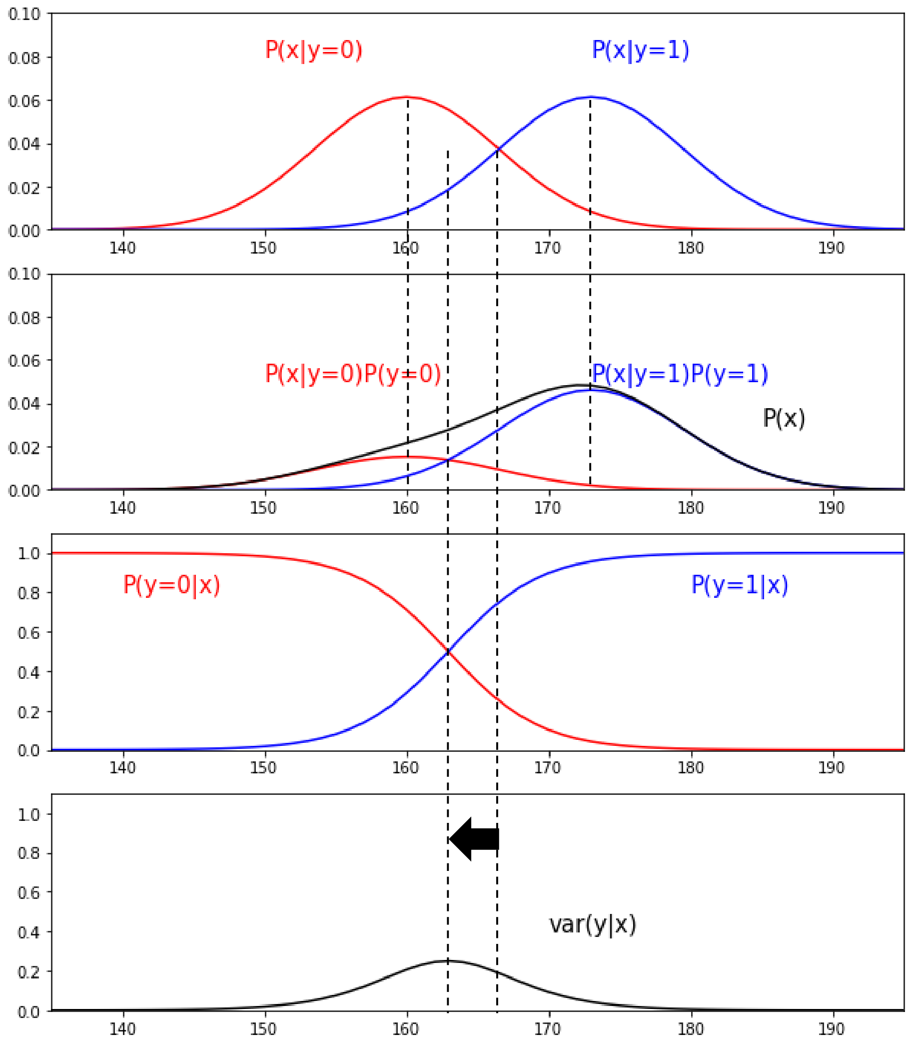
Equal Variance and Equal Prior



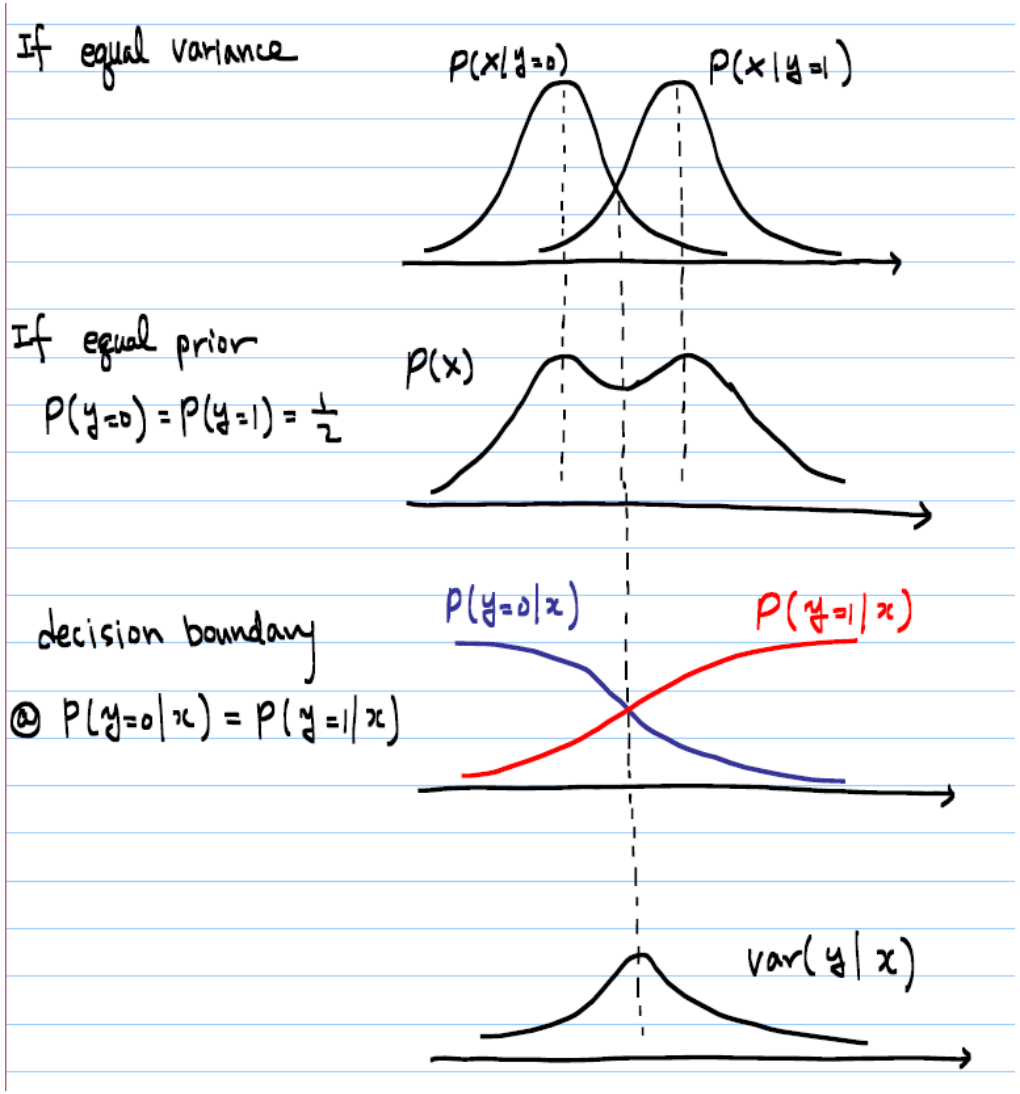


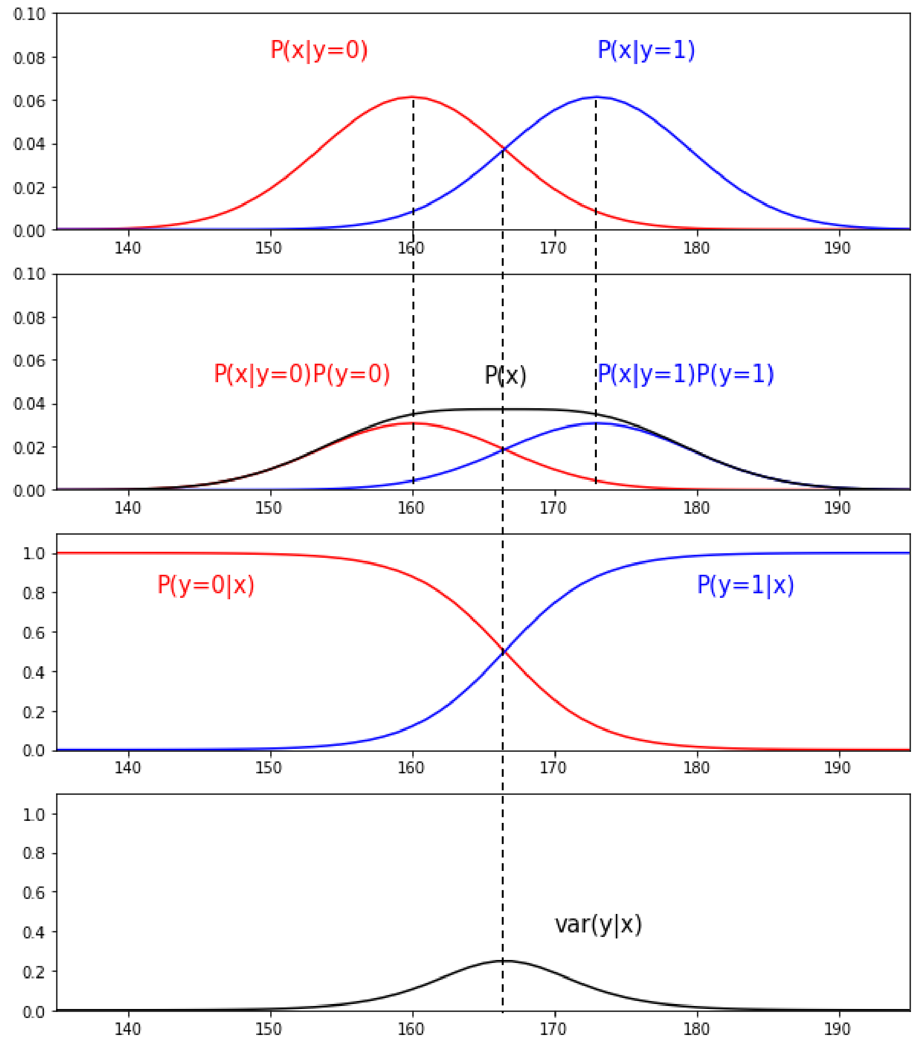
Equal Variance and Not Equal Prior



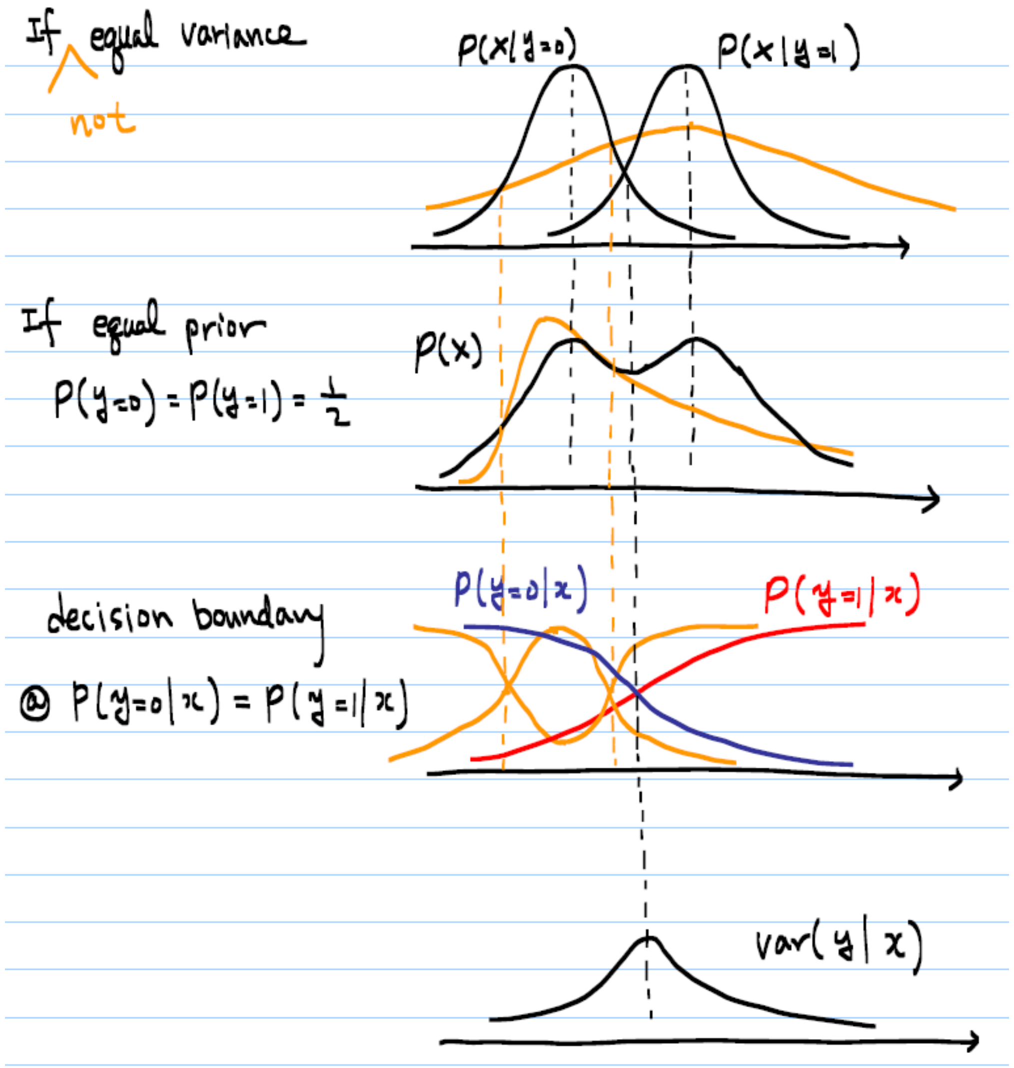


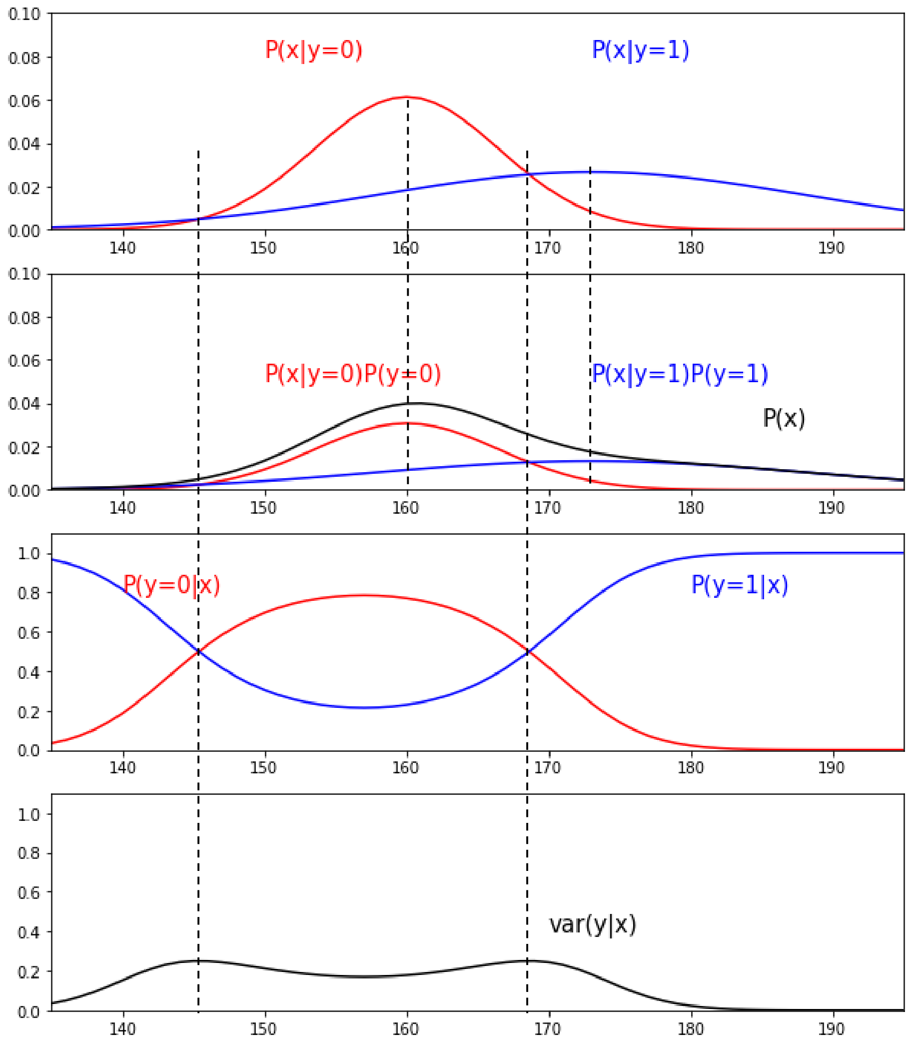
Equal Variance and Equal Prior





Not Equal Variance and Equal Prior





Back to Logistic Regression

- Logistic regression makes assumption on the posterior

$$P(y \mid x, \omega) = \sigma\left(y\omega^T x\right) = \frac{1}{1 + \exp(-y\omega^T x)}$$

- At the decision boundary labels $-1/+1$ becomes equiprobable

$$P(y = +1 \mid x, \omega) = P(y = -1 \mid x, \omega)$$

$$\frac{1}{1 + \exp(-\omega^T x)} = \frac{1}{1 + \exp(\omega^T x)}$$

$$\exp\left(-\omega^T x\right) = \exp\left(\omega^T x\right)$$

$$\omega^T x = 0$$

Probability Density Estimation:

Kernel Density Estimation

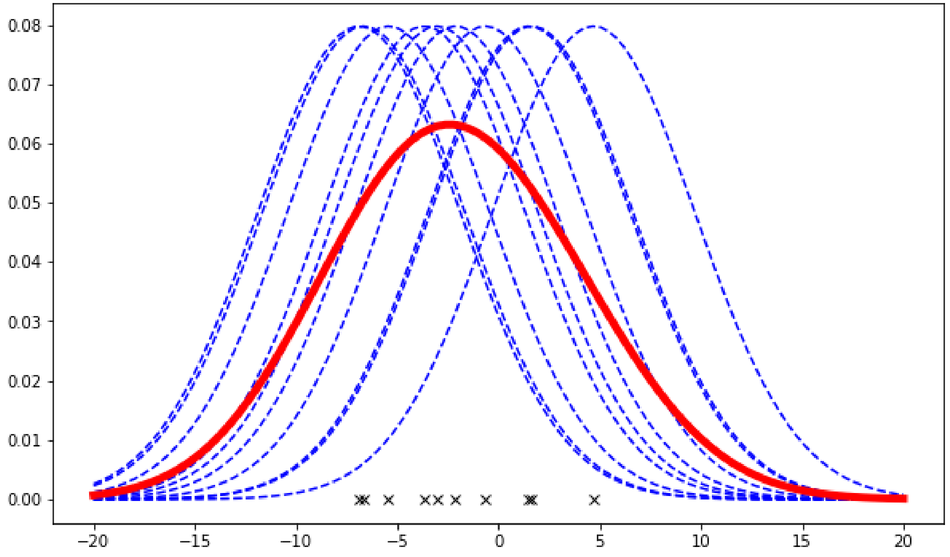
Kernel Density Estimation

- *non-parametric* estimate of density
- Lecture: Learning Theory (Reza Shadmehr, Johns Hopkins University)

https://www.youtube.com/watch?v=a357NoXy4Nk&feature=emb_logo

Kernel Density Estimation

```
m = 10
mu = 0
sigma = 5

x = np.random.normal(mu,sigma,[m,1])
xp = np.linspace(-20,20,100)
y0 = np.zeros([m,1])

X = []

for i in range(m):
    X.append(norm.pdf(xp,x[i,0],sigma))

X = np.array(X).T
Xnorm = np.sum(X,1)/m

plt.figure(figsize=(10,6))
plt.plot(x,y0,'kx')
plt.plot(xp,X,'b--')
plt.plot(xp,Xnorm,'r',linewidth=5)
plt.show()
```



Probability Density Estimation:

Bayesian Density Estimation

Bayesian Density Estimation

- Not parameter estimation any more
- Probability density estimation
  - (Gaussian case: parameter = pdf)

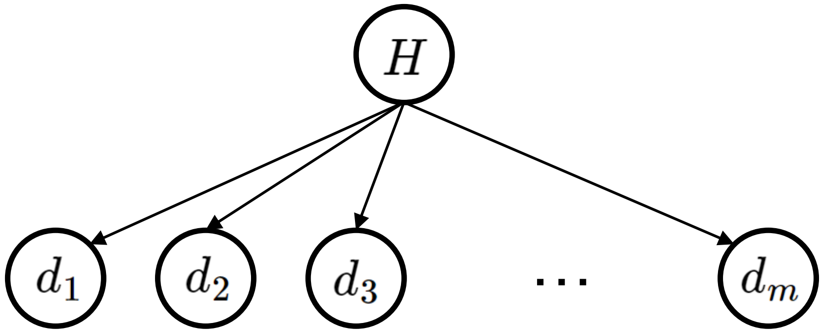
- Start with prior beliefs, which can be thought of as a summary of opinions.
  - might be subjective

- Given our prior, we can update our opinion, and produce a new opinion.
  - This new distribution is called the posterior

- Iterate
  - if more data is available

Hidden State

- Estimate a probability density function of a hidden state from multiple observations

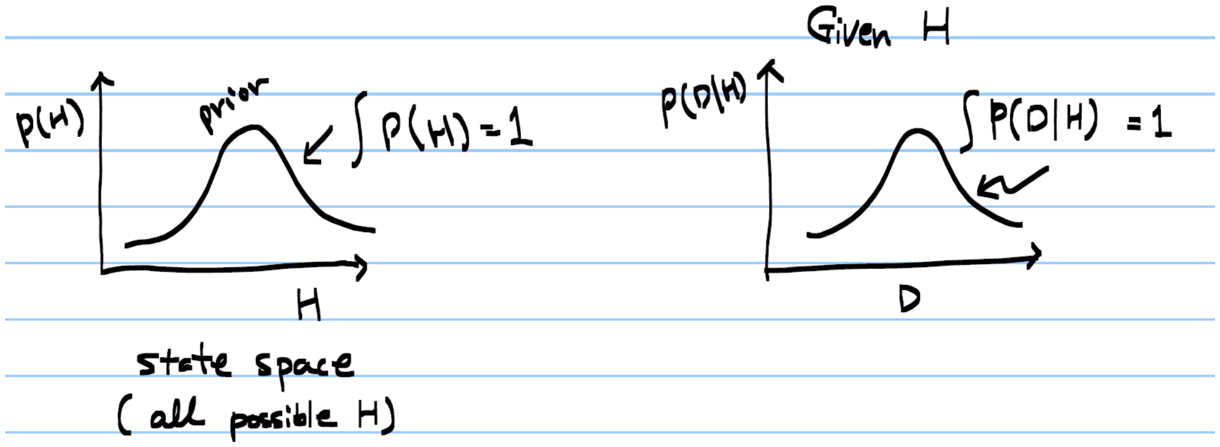


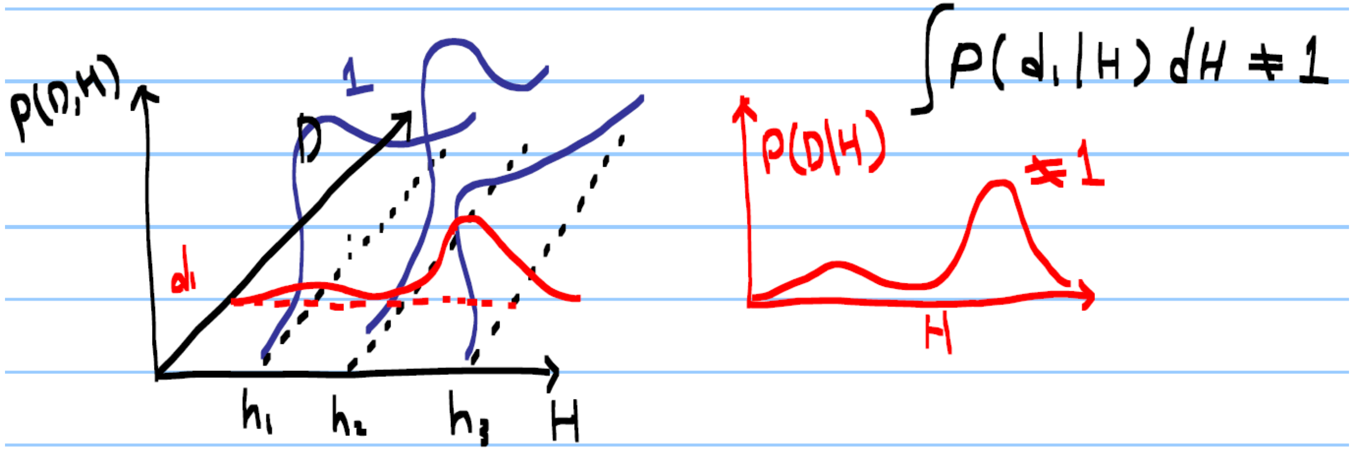
- H: Hypothesis, hidden state
- $D = \{d_1, d_2, \cdots, d_m\}$: data, observation, evidence

$$P(H \mid D) = \frac{P(D \mid H)P(H)}{P(D)}$$

Likelihood

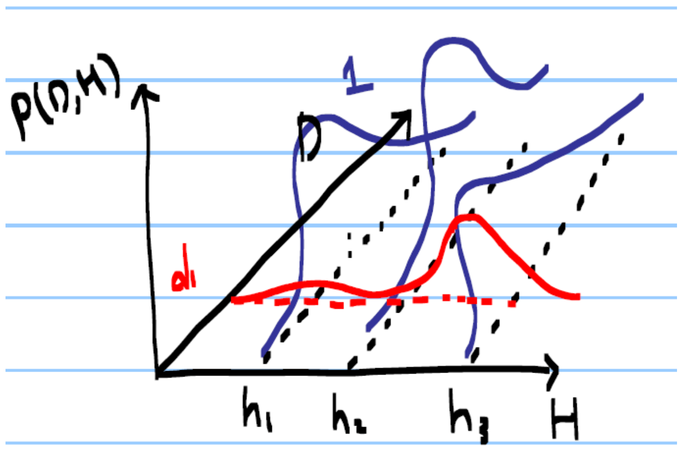
$$P(H \mid D) = \frac{P(D \mid H)P(H)}{P(D)}$$

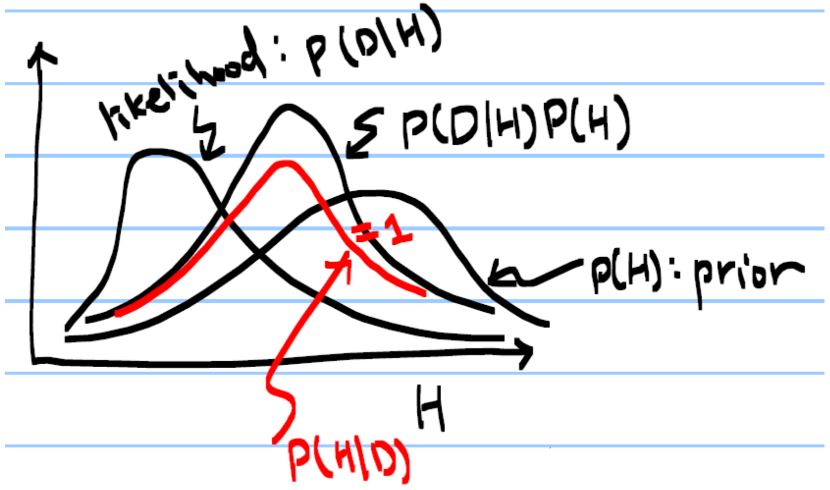




Posterior

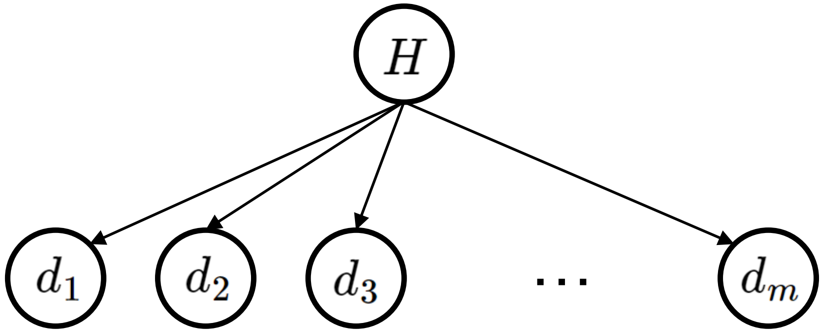
$$P(H \mid D) = \frac{P(D \mid H)P(H)}{P(D)}$$





Combining Multiple Evidences

- Compute posterior probability
- Assume conditional independence



$$P(H \mid \underbrace{d_1, d_2, \cdots, d_m}_{\text{multiple evidences}}) = \frac{P(d_1, d_2, \cdots, d_m \mid H)\ P(H)}{P(d_1, d_2, \cdots, d_m)}$$

$$= \frac{P(d_1 \mid H)P(d_2 \mid H) \cdots P(d_m \mid H)\ P(H)}{P(d_1, d_2, \cdots, d_m)}$$

$$= \eta \prod_{i=1}^{m} P(d_i \mid H)P(H), \qquad \eta : \text{normalizing}$$

Recursive Bayesian Estimation

- Two identities

$$P(a, b) = P(a \mid b)P(b)$$
$$P(a, b \mid c) = P(a \mid b, c)P(b \mid c)$$

- When multiple $d_1, d_2, \cdots$

$$P(H \mid d_1) = \frac{P(d_1 \mid H)P(H)}{P(d_1)} = \eta_1 \, P(d_1 \mid H)\underbrace{P(H)}_{\text{prior}}$$

$$P(H \mid d_1 d_2) = \frac{P(d_1 d_2 \mid H)P(H)}{P(d_1 d_2)} = \frac{P(d_1 \mid d_2, H)P(d_2 \mid H)P(H)}{P(d_1 d_2)} = \frac{P(d_1 \mid H)P(d_2 \mid H)P(H)}{P(d_1 d_2)} = \eta_2 \, P(d_2 \mid H) \, \underbrace{P(H \mid d_1)}_{\text{acting as a prior}}$$

$$\vdots$$

$$P(H \mid d_1, d_2, \cdots, d_m) = \eta_m \, P(d_m \mid H)\underbrace{P(H \mid d_1, d_2, \cdots, d_{m-1})}_{\text{acting as a prior}}$$

Recursive Bayesian Estimation

- Recursive

$$P_0(H) = P(H) \implies P(H \mid d_1) = P_1(H) \implies P(H \mid d_1 d_2) = P_2(H) \implies \cdots$$

- Recursive Bayesian Estimation

$$P(H \mid D) = \frac{P(D \mid H)P(H)}{P(D)}$$

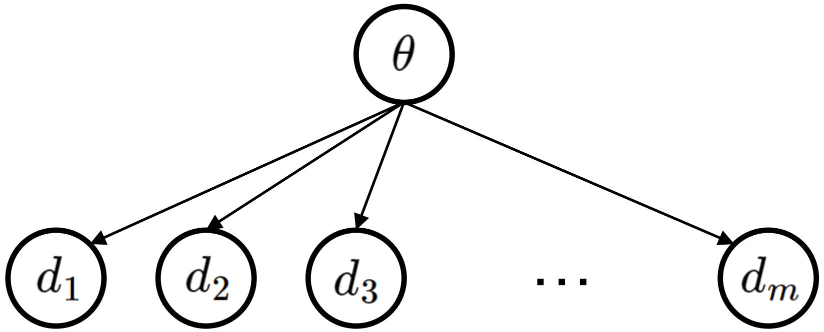
Posterior → prior as more evidence is collected

POSTECH

Example 1: Bernoulli Model

$$d = \{0, 1\}, \qquad \theta \in [0, 1]$$

$$p(d \mid \theta) = P[D = d \mid \theta] = \theta^d (1-\theta)^{1-d} = \begin{cases} 1-\theta, & d = 0 \\ \theta, & d = 1 \end{cases}$$



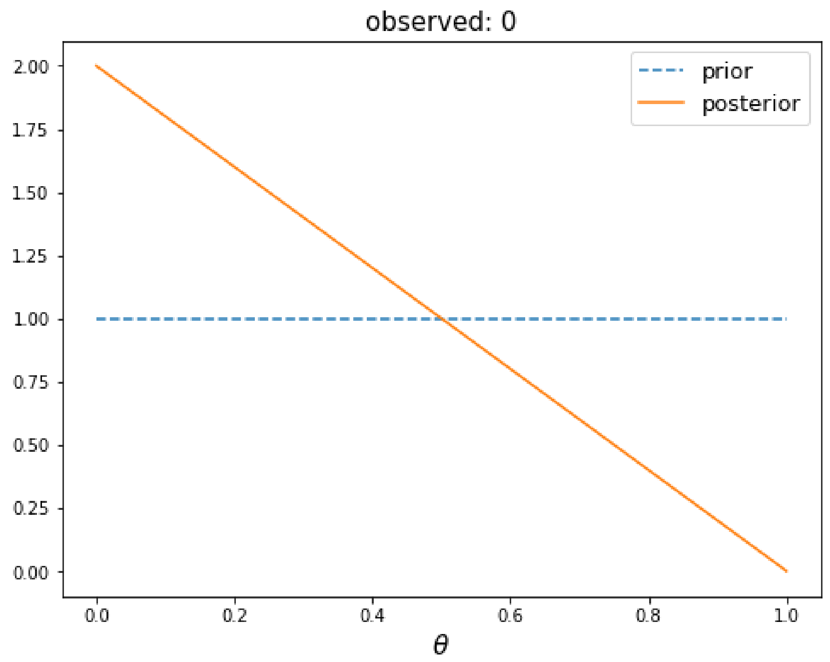
Bernoulli Model

$$d = \{0, 1\}, \qquad \theta \in [0, 1]$$

$$p(d \mid \theta) = P[D = d \mid \theta] = \theta^d (1-\theta)^{1-d} = \begin{cases} 1-\theta, & d = 0 \\ \theta, & d = 1 \end{cases}$$

```python
def normalize(y, x):
    return y / np.trapz(y, x)
```

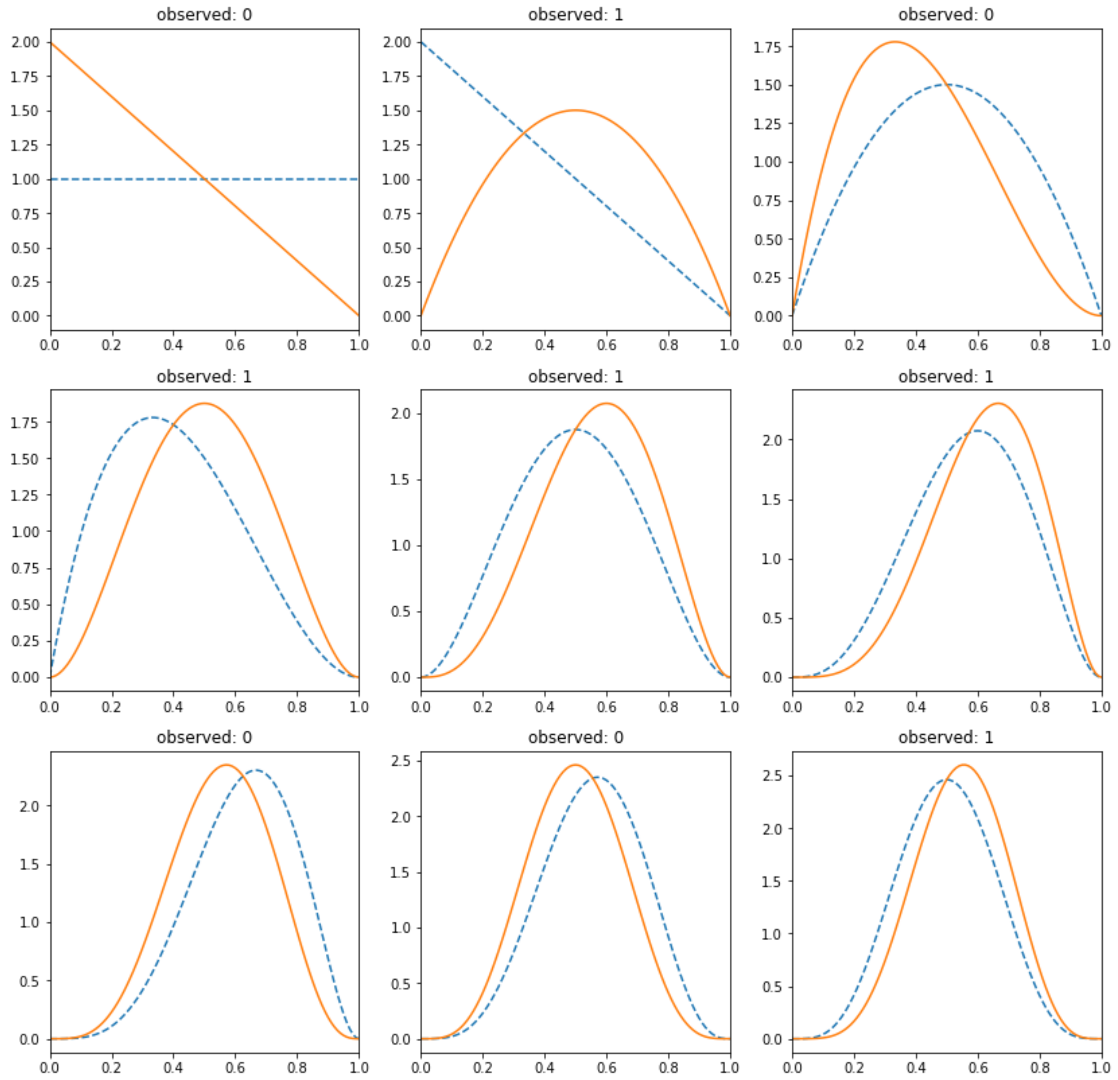
```python
N = 101

theta = np.linspace(0, 1, N)
prior = normalize(np.repeat(1, N), theta)

d = np.random.choice([0,1])

likelihood = theta**d * (1 - theta)**(1 - d)

posterior = likelihood * prior
posterior = normalize(posterior, theta)
```



Recursive Bayesian Estimation

```python
def bernoulli_model(d, theta, prior):
    likelihood = theta**d * (1 - theta)**(1 - d)
    posterior = likelihood * prior
    return normalize(posterior, theta)
```

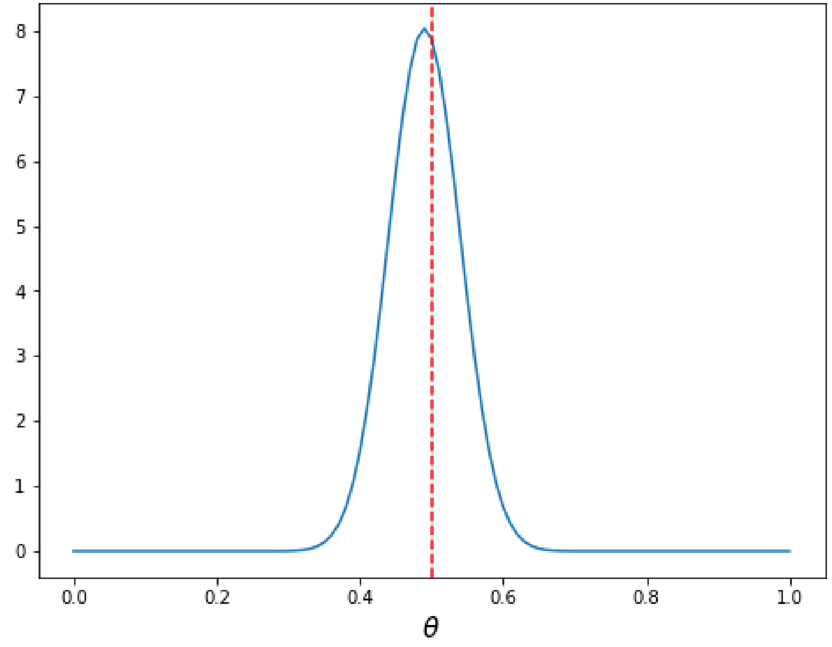
```python
for n in range(9):
    observed = np.random.choice([0,1])
    posterior = bernoulli_model(observed, theta, prior)

    ax[n].plot(theta, prior, linestyle = '--')
    ax[n].plot(theta, posterior)
    ax[n].set_title('observed: %d' % observed)
    ax[n].set_xlim([0,1])

    prior = posterior
```



Recursive Bayesian Estimation

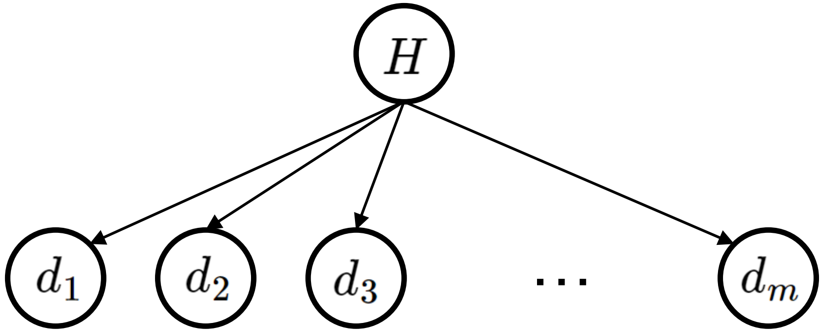
```
prior = normalize(np.repeat(1, N), theta)

observation = []
for _ in range(100):
    observed = np.random.choice([0,1])
    observation.append(observed)
    posterior = bernoulli_model(observed, theta, prior)

    prior = posterior

print(observation,'\n')
print(np.mean(observation))

plt.figure(figsize = (8,6))
plt.plot(theta, posterior)
plt.axvline(0.5, color = 'red', linestyle = '--')
plt.xlabel(r'$\theta$', fontsize = 15)
plt.show()
```

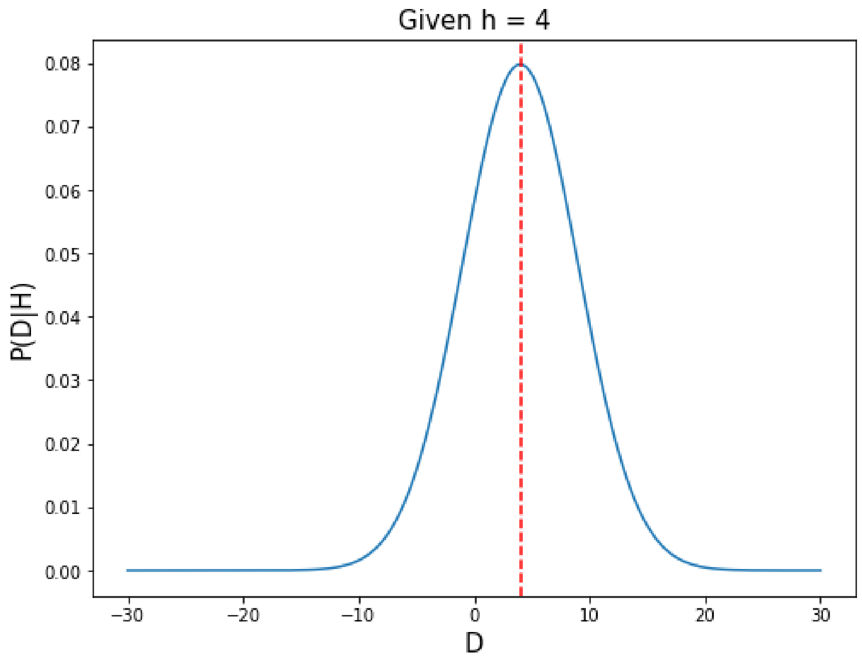
```
0.49
```



Example 2: Gaussian Model

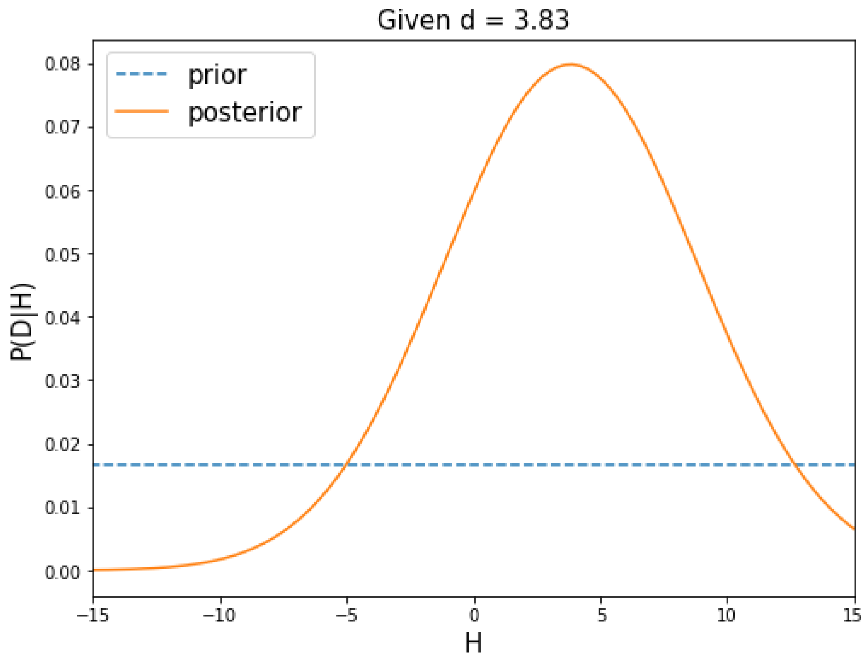
$$p(d \mid h) \sim \mathcal{N}(h, \sigma^2)$$





Posterior Probability

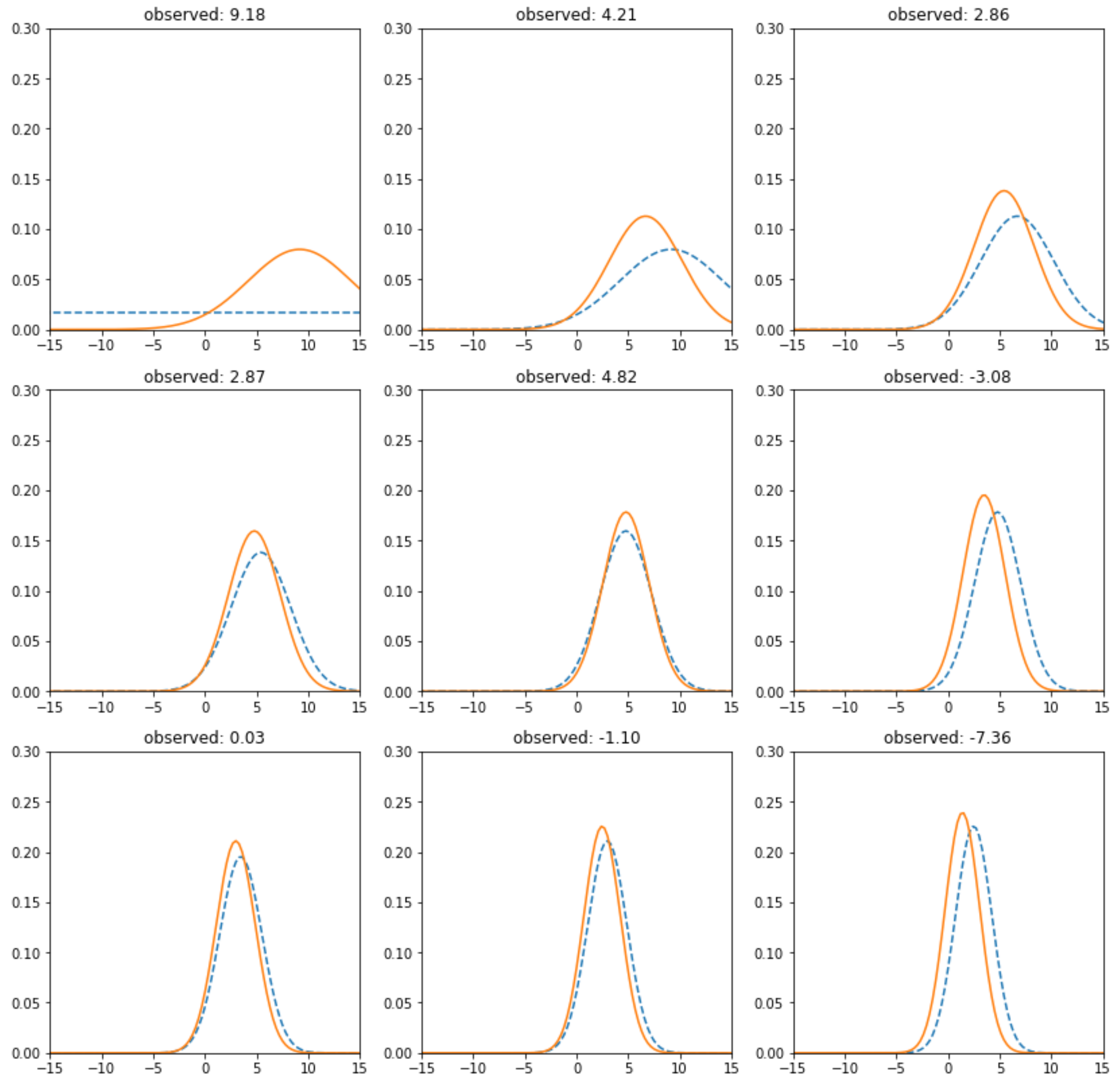
```
H = np.linspace(-30,30, N)
prior = normalize(np.repeat(1, N), H)

d = np.random.normal(0, sigma)

likelihood = []
for h in H:
    likelihood.append(stats.norm.pdf(d,h,sigma))

posterior = likelihood * prior
posterior = normalize(posterior, H)
```



Recursive Bayesian Estimation

```python
def Gaussian_model(d, H, prior):
    likelihood = []
    for h in H:
        likelihood.append(stats.norm.pdf(d,h,sigma))

    posterior = likelihood * prior
    return normalize(posterior, H)
```

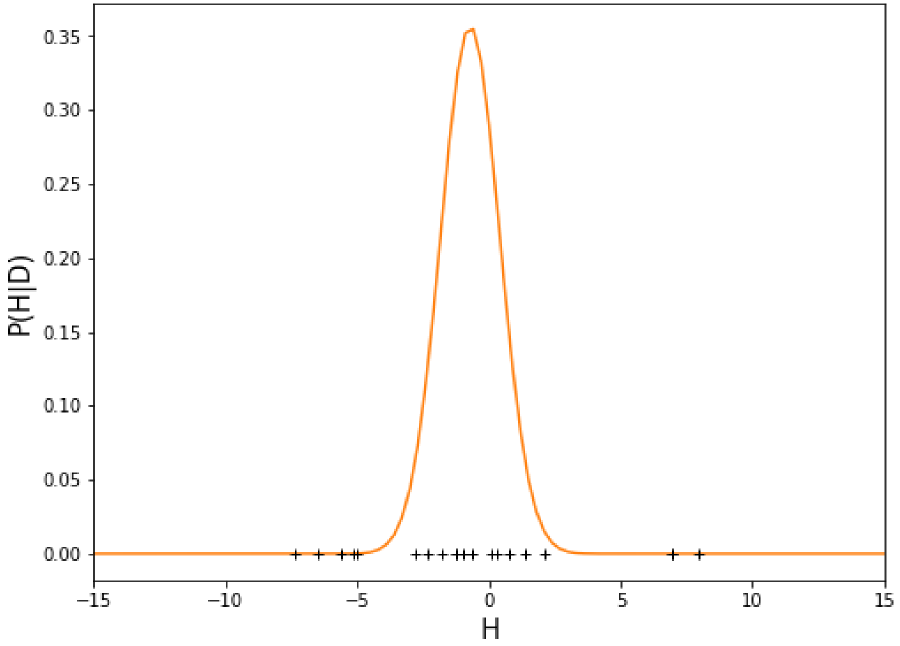
```python
fig, ax = plt.subplots(ncols = 3, nrows = 3, figsize = (14,14))
ax = np.ravel(ax)

for n in range(9):
    observed = np.random.normal(0, sigma)
    posterior = Gaussian_model(observed, H, prior)

    ax[n].plot(H, prior, '--')
    ax[n].plot(H, posterior)
    ax[n].set_title('observed: %1.2f' % observed)
    ax[n].set_ylim([0,0.3])
    ax[n].set_xlim([-15,15])

    prior = posterior

plt.show()
```



Recursive Bayesian Estimation

```
prior = normalize(np.repeat(1, N), H)

observation = []

for _ in range(20):
    d = np.random.normal(0, sigma)
    observation.append(d)
    posterior = Gaussian_model(d, H, prior)

    prior = posterior
```

```
-0.7185173390571822
```



Summary

- Bayesian Machine Learning

- Bayesian Classifier

- Bayesian Density Estimation

- Bayes' Rule
  - Prior
  - Likelihood
  - Posterior