module_1_oa1

April 10, 2025

P(A|B) = (P(B|A) * P(A)) / P(B) P(B) = (P(B|A) * P(A)) / P(A|B) = P(B|A) * P(A) + P(B| not A) * P(not A)

A = infected B = pos test

```
        (Start)
        /     \
   Disease    No Disease
   (0.05)       (0.95)
   /    \       /    \
```

+Test -Test +Test -Test (0.99) (0.01) (0.01) (0.99)

sens P(B|A) = TP/(TP+FN) = P(positive test | infected) = 0.99 spec P(not B | not A) = TN/(TN+FP) = P(negative test | not infected) = 0.99 P(infected) = P(A) = 0.05

P(not infected) = 1 - P(infected) = 0.95 P(pos test | not infected) = 1 - spec = 0.01 P(neg test | infected) = 1 - sens = 0.01

P(pos test) = P(pos test | infected) * P(infected) + P(pos test | not infected) * P(not infeccted) P(pos test) = 0.99 * 0.05 + 0.01 * 0.95 = 0.059

[3]:
```
import plotly.graph_objects as go
import numpy as np


def calculate_prob(sens, spec, infect):
    p_b_not_a = 1 - spec  # False positive rate
    p_not_a = 1 - infect  # Probability of not being infected
    p_b = sens * infect + p_b_not_a * p_not_a  # Total probability of a␣
 ↪positive test
    p_a_b = (sens * infect) / p_b  # Posterior probability of being infected␣
 ↪given a positive test
    return p_a_b


infection_rates = np.linspace(0,0.5,100)
sensitivities = [0.6, 0.8, 0.9, 0.995, 0.999]
initial_specificity = 0.99

fig = go.Figure()
```

```
traces = []
for sens in sensitivities:
    post_probs = [calculate_prob(sens, initial_specificity, inf) for inf in␣
 ↪infection_rates]
    traces.append(go.Scatter(x=infection_rates,
                             y=post_probs,
                             name=f"Sensitivity = {sens}",
                             mode='lines',
                             line=dict(shape='spline'),
                             hovertemplate=
                                "Infection Rate: %{x:.2f}<br>" +  # X-axis␣
 ↪value (infection rate)
                                "Posterior Probability: %{y:.4f}<br>" +  #␣
 ↪Y-axis value (posterior probability)
                                "Sensitivity: " + str(sens) + "<br>" +  #␣
 ↪Sensitivity value
                                "<extra></extra>"  # Remove trace name from the␣
 ↪hovertext
            ))
fig.add_traces(traces)

steps = []
for spec in np.arange(0.8, 1.0, 0.001):
    step = dict(
        method="update",
        args=[{"type": "scatter", "x": [infection_rates] * len(sensitivities),␣
 ↪"y": [[calculate_prob(sens, spec, inf) for inf in infection_rates] for sens␣
 ↪in sensitivities]}],
        label=f"{spec:.3f}",  # Label for the slider step
    )
    steps.append(step)

fig.update_layout(
    title='Posterior Probability vs Infection Rate for Different Sensitivities',
    xaxis={'title':{'text':'Infection rate'}},
    yaxis={'title':{'text':'Posterior Probability being infected (P(A|B))'}},
    sliders=[dict(
            active=190,  # Default slider position (for specificity 0.99)
            currentvalue={"prefix": "Specificity: "},
            pad={"t": 50},
            steps=steps
        )])

fig.show()
```

population = 10'000 infected = 5% sensitivity = 99% specificity = 99%

infected population = 500

TP = 0.99 * 500 = 495 FN = 5 TN = 0.99 * 9'500 = 9405 FP = 95

P(A|B) = P(positive test|infected) = TP/(TP+FP) = 495/(495+95) = 495/590 = 83.9%