Gaussian Process Classification - Detailed Hand Calculations

1. Model Setup and Notation

Consider a simple example with N=3 training points:

- Training inputs: X = [[0], [1], [2]] (1D inputs)
- Training labels: y = [0, 1, 0]
- Kernel: Squared exponential $k(x,x') = \kappa \exp(-|x-x'|^2/(2\ell^2))$
- Hyperparameters: $\kappa = 1$, $\ell = 1$

2. Prior Calculations

2.1 Kernel Matrix K

Compute $K_{ij} = k(x_i, x_j) = \exp(-|x_i - x_j|^2/2)$:

```
K_11 = exp(-|0-0|²/2) = exp(0) = 1.000
K_12 = exp(-|0-1|²/2) = exp(-0.5) = 0.607
K_13 = exp(-|0-2|²/2) = exp(-2) = 0.135
K_22 = exp(-|1-1|²/2) = exp(0) = 1.000
K_23 = exp(-|1-2|²/2) = exp(-0.5) = 0.607
K_33 = exp(-|2-2|²/2) = exp(0) = 1.000
```

Kernel matrix:

```
K = [[1.000, 0.607, 0.135],
     [0.607, 1.000, 0.607],
     [0.135, 0.607, 1.000]]
```

2.2 Cholesky Decomposition

Compute L such that $K = LL^T$:

```
L = [[1.000, 0.000, 0.000],
     [0.607, 0.795, 0.000],
     [0.135, 0.692, 0.709]]
```

2.3 Log Determinant

$\log|K| = 2 * \Sigma \log(L_{ii}) = 2 * (\log(1) + \log(0.795) + \log(0.709)) = 2 * (0 - 0.229 - 0.343) = -1.144$

3. MAP Estimation Calculations

3.1 Log Joint Distribution

log p(y,f) = log p(y|f) + log p(f)

Log prior:

```
log p(f) = -N/2 log(2π) - 1/2 log|K| - 1/2 f^T K^{-1} f
         = -3/2 log(2π) - 1/2(-1.144) - 1/2 f^T K^{-1} f
         = -2.76 + 0.572 - 1/2 f^T K^{-1} f
```

Log likelihood:

```
log p(y|f) = Σ[y_i log σ(f_i) + (1-y_i) log(1-σ(f_i))]
           = 0·log σ(f_1) + 1·log(1-σ(f_1)) +
             1·log σ(f_2) + 0·log(1-σ(f_2)) +
             0·log σ(f_3) + 1·log(1-σ(f_3))
           = log(1-σ(f_1)) + log σ(f_2) + log(1-σ(f_3))
```

3.2 Gradient Computation

∇_f log p(y,f) = (y - σ(f)) - K^{-1}f

For f = [f_1, f_2, f_3]:

```
Gradient from likelihood:
g_1 = 0 - σ(f_1) = -σ(f_1)
g_2 = 1 - σ(f_2) = 1 - σ(f_2)
g_3 = 0 - σ(f_3) = -σ(f_3)

Gradient from prior: -K^{-1}f
```

3.3 Iterative Optimization

Starting from f^(0) = [0, 0, 0]:

Iteration 1:

```
σ(0) = 0.5 for all components
g_lik = [0-0.5, 1-0.5, 0-0.5] = [-0.5, 0.5, -0.5]
g_prior = -K^{-1}[0,0,0] = [0, 0, 0]
∇ = [-0.5, 0.5, -0.5]
f^(1) = f^(0) + α∇ (with appropriate step size α)
```

Continue iterations until convergence: Final MAP estimate: f_MAP ≈ [-0.8, 0.9, -0.8]

4. Hessian and Posterior Covariance

4.1 Hessian of Log Likelihood

At f_MAP, compute $\Lambda = -\nabla^2 \log p(y|f)$:

```
Λ_11 = σ(f_1)(1-σ(f_1)) = σ(-0.8)(1-σ(-0.8))
     = 0.31 × 0.69 = 0.214
Λ_22 = σ(f_2)(1-σ(f_2)) = σ(0.9)(1-σ(0.9))
     = 0.71 × 0.29 = 0.206
Λ_33 = σ(f_3)(1-σ(f_3)) = σ(-0.8)(1-σ(-0.8))
     = 0.31 × 0.69 = 0.214
```

Lambda matrix:

```
Λ = [[0.214, 0.000, 0.000],
     [0.000, 0.206, 0.000],
     [0.000, 0.000, 0.214]]
```

4.2 Posterior Covariance via Woodbury

S = (K^{-1} + Λ)^{-1} = K - K Λ^{1/2} (I + Λ^{1/2} K Λ^{1/2})^{-1} Λ^{1/2} K

Step 1: Compute Λ^{1/2}

```
Λ^{1/2} = [[0.463, 0.000, 0.000],
           [0.000, 0.454, 0.000],
           [0.000, 0.000, 0.463]]
```

Step 2: Compute B = I + Λ^{1/2} K Λ^{1/2}

```
Λ^{1/2} K Λ^{1/2} = [[0.214, 0.128, 0.029],
                     [0.125, 0.206, 0.125],
                     [0.029, 0.128, 0.214]]

B = [[1.214, 0.128, 0.029],
     [0.125, 1.206, 0.125],
     [0.029, 0.128, 1.214]]
```

Step 3: Solve systems using Cholesky of B

```
L_B = cholesky(B)
e = solve(L_B, Λ^{1/2} K)
S = K - e^T e
```

Final posterior covariance S (approximate):

```
S ≈ [[0.744, 0.475, 0.106],
     [0.475, 0.767, 0.475],
     [0.106, 0.475, 0.744]]
```

5. Prediction Calculations

5.1 Predict at x* = 1.5

Step 1: Compute covariances

```
k(x*, x_1) = exp(-|1.5-0|²/2) = exp(-1.125) = 0.325
k(x*, x_2) = exp(-|1.5-1|²/2) = exp(-0.125) = 0.882
k(x*, x_3) = exp(-|1.5-2|²/2) = exp(-0.125) = 0.882
k(x*, x*) = exp(0) = 1.000
```

Covariance vector: k* = [0.325, 0.882, 0.882]

Step 2: Compute K^{-1}m

First, compute K^{-1}:

```
K^{-1} = [[ 1.352, -0.797, -0.081],
          [-0.797,  1.784, -0.797],
          [-0.081, -0.797,  1.352]]
```

Then: K^{-1}m = K^{-1}[-0.8, 0.9, -0.8]

```
= [[-1.082 + 0.717 + 0.065],
   [ 0.638 + 1.606 + 0.638],
   [ 0.065 + 0.717 - 1.082]]
= [-0.300, 2.882, -0.300]
```

Step 3: Predictive mean

```
μ* = k*^T K^{-1} m
   = [0.325, 0.882, 0.882] · [-0.300, 2.882, -0.300]
   = -0.098 + 2.542 - 0.265
   = 2.179
```

Step 4: Predictive variance

Compute K^{-1}k*:

```
K^{-1}k* = [[ 0.440 - 0.703 - 0.071],
            [-0.259 + 1.573 - 0.703],
            [-0.026 - 0.703 + 1.193]]
         = [-0.334, 0.611, 0.464]
```

Then: k*^T K^{-1} (K-S) K^{-1} k*

First compute K-S:

```
K-S ≈ [[0.256, 0.132, 0.029],
       [0.132, 0.233, 0.132],
       [0.029, 0.132, 0.256]]
```

Finally:

```
σ*² = k** - k*^T K^{-1} (K-S) K^{-1} k*
    = 1.000 - 0.082
    = 0.918
```

5.2 Class Probability Prediction

Using probit approximation:

```
p(y*=1) = Φ(μ* / √(8/π + σ*²))
        = Φ(2.179 / √(2.546 + 0.918))
        = Φ(2.179 / √3.464)
        = Φ(2.179 / 1.861)
        = Φ(1.171)
        ≈ 0.879
```

6. Complete Workflow Summary

Input

- X = [[0], [1], [2]]
- y = [0, 1, 0]
- Test point: x* = 1.5

Laplace Approximation

1. Compute kernel matrix K
2. Find MAP: f_MAP ≈ [-0.8, 0.9, -0.8]
3. Compute Hessian: Λ = diag([0.214, 0.206, 0.214])
4. Posterior covariance: S via Woodbury identity

Prediction

1. Mean: μ* = 2.179
2. Variance: σ^{*2} = 0.918
3. Probability: p(y*=1) ≈ 0.879

Interpretation

- High probability (87.9%) of class 1 at x* = 1.5
- Reasonable uncertainty (σ ≈ 0.96)
- Smooth interpolation between training points

7. Key Mathematical Insights

1. **MAP estimation**: Balances likelihood and prior
2. **Laplace approximation**: Gaussian around mode
3. **Woodbury identity**: Numerical stability
4. **Probit trick**: Closed-form integration

5. **Uncertainty propagation**: From f* to y*

This detailed calculation demonstrates how GPC combines flexibility of Gaussian Processes with requirements of classification through careful approximations.