

## Decisions

**Ludwig Krippahl**

## Summary

- Bayesian Learning
- Maximum Likelihood vs Maximom A Posteriori
- Monte Carlo and computing prior probability distributions
- Decisions and costs

## Bayesian Learning

## Bayesian vs Frequentist probabilities

- To find parameters in some cases (E.g. regression, logistic regression) we maximized the likelihood:

$$\hat{\theta}_{ML} = \arg \max_{\theta} \prod_{t=1}^n p(x^t, y^t)$$

- Rewriting as conditional probabilities, and since  $p(x^t)$  is constant:

$$\prod_{t=1}^n p(x^t, y^t) = \prod_{t=1}^n p(y^t | x^t) \times \prod_{t=1}^n p(x^t) \quad \hat{\theta}_{ML} = \arg \max_{\theta} \prod_{t=1}^n p(y^t | x^t; \theta)$$

- Under a frequentist interpretation, probability is the frequency in the limit of infinite trials.
- So  $\theta$  is unknown but not a random variable.

## Bayesian vs Frequentist probabilities

- Under a bayesian interpretation, probability is a measure of knowledge and uncertainty and  $\theta$  can be seen as another random variable with its own probability distribution

- Given **prior**  $p(\theta)$  and sample  $S$ , update **posterior**  $p(\theta|S)$ :

$$p(\theta|S) = \frac{p(S|\theta)p(\theta)}{p(S)}$$

- where  $p(S)$  is the marginal probability of  $S$  (the **evidence**) and  $p(S|\theta)$  is the **likelihood** of  $\theta$

$$p(\theta|S) = \frac{p(S|\theta)p(\theta)}{p(S)} \Leftrightarrow p(\theta|S) = \frac{\prod_{t=1}^n p(y^t|x^t, \theta)p(\theta)}{p(S)}$$

## Bayesian vs Frequentist probabilities

- Since  $p(S)$  is generally unknown and constant, we approximate the posterior with the **Maximum A Posteriori** (MAP) estimate:

$$\hat{\theta}_{MAP} = \arg \max_{\theta} \prod_{t=1}^n p(y^t | x^t, \theta) p(\theta)$$

- ML and MAP are similar but with a significant difference:

$$\hat{\theta}_{ML} = \arg \max_{\theta} \prod_{t=1}^n p(y^t | x^t; \theta)$$

- Treating the parameters as a probability distribution leads naturally to regularization due to the inclusion of the prior probability distribution of the parameters  $p(\theta)$
- (e.g. Bayesian logistic regression)

## Computing priors

- **Uninformative Priors**: the prior probability has little impact on the posterior, and MAP becomes similar to ML
- In some cases, a uniform distribution can suffice.
- In other cases, we need different distributions. E.g. line slope on linear regression
- We may also want to include prior information about the parameters
- Often results in probability distributions for which we have no analytical expression for expected values
- Bayesian learning generally requires numerical sampling methods (Monte Carlo), which can make it computationally more demanding
- but we can explicitly use prior probability distributions instead of ad-hoc regularization

## Decisions and costs



# Decisions and costs

## Measuring error

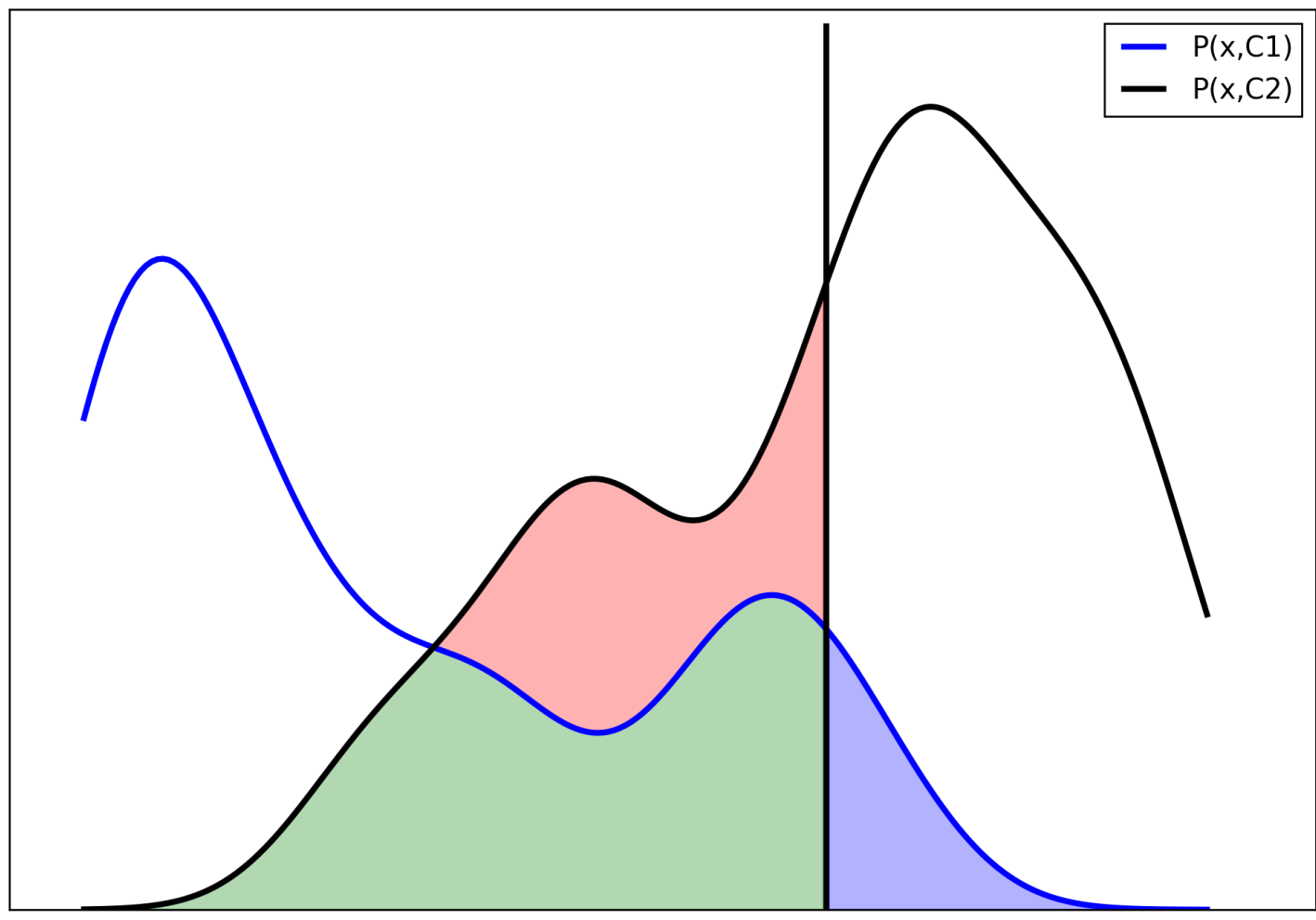
- So far, the loss functions we used were all measures of error
- But sometimes, the error may not be the best loss function

## Loss functions

- Suppose we have the joint probability distributions  $P(x, C_1)$  and  $P(x, C_2)$
- We also have a classifier that classifies an example as  $C_2$  if  $x > \hat{x}$  or  $C_1$  otherwise

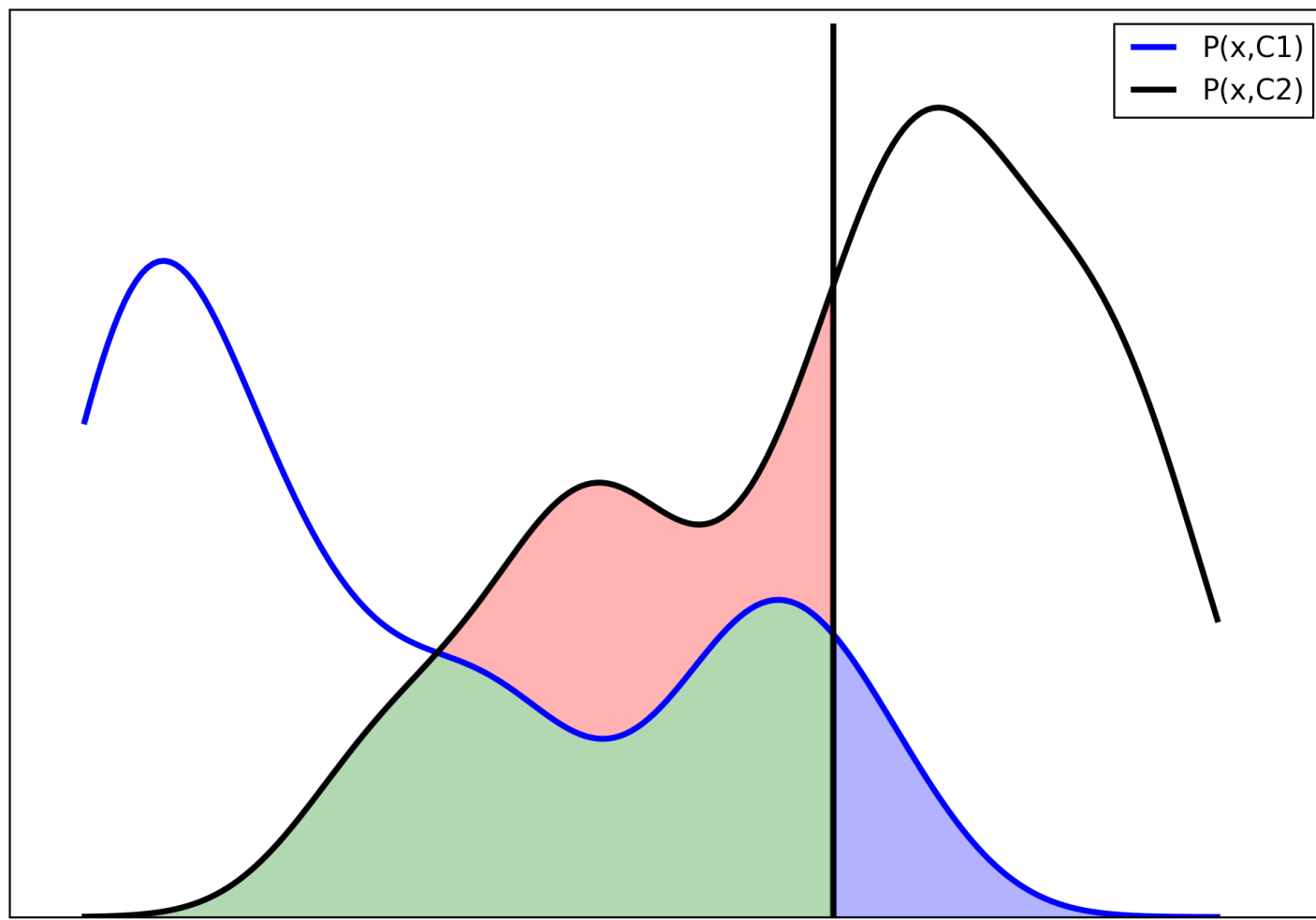
# Decisions and costs

- Errors depend on the choice of  $\hat{x}$



# Decisions and costs

- Red and green:  $C_2$  misclassified; Blue:  $C_1$  misclassified



# Decisions and costs

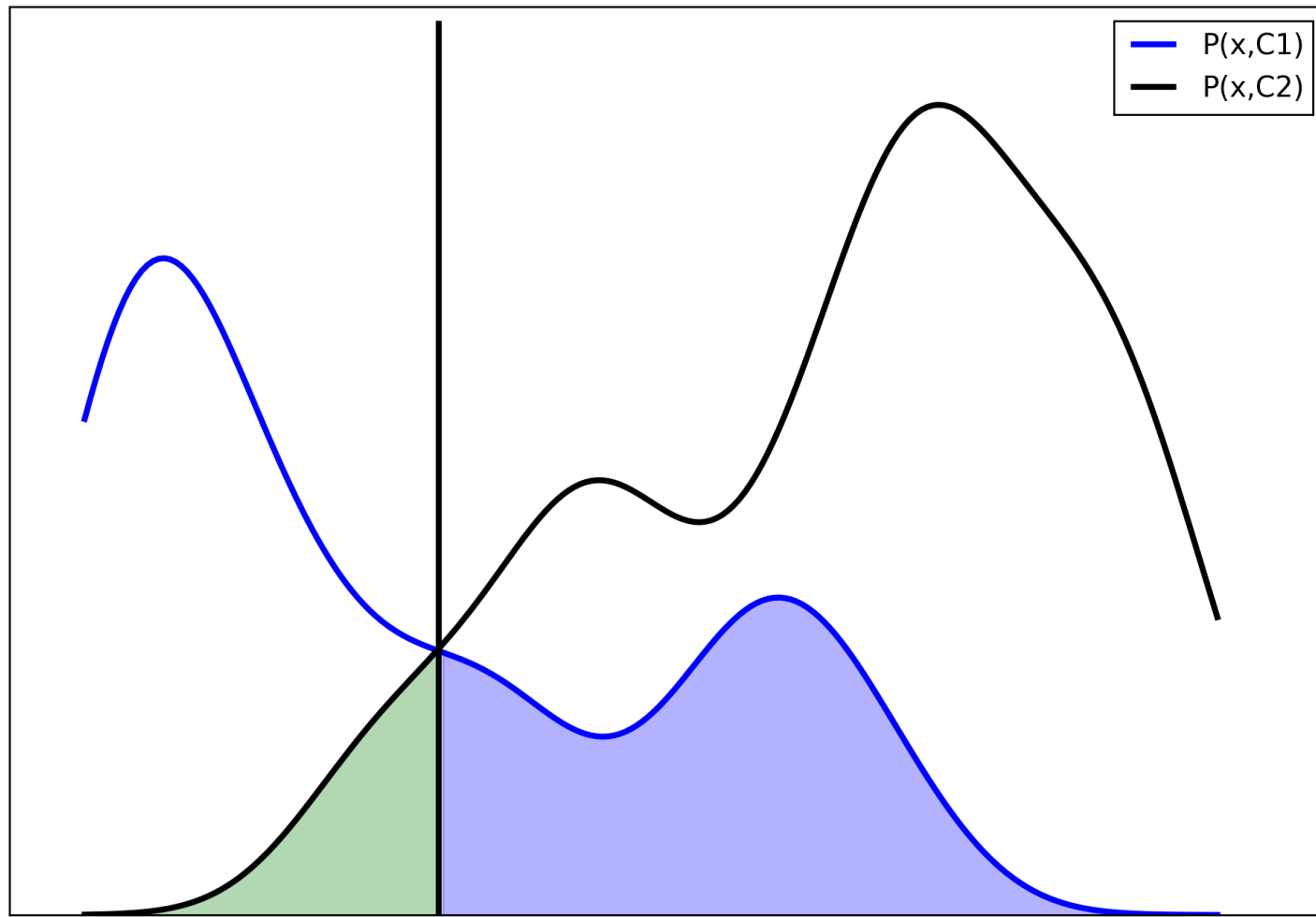
- Minimizing the misclassification rate is equivalent to maximizing the probability of  $x$  corresponding to the predicted class
- This can be done by choosing  $\hat{x}$  such that

$$P(C_1|x) > P(C_2|x) \text{ for } x < \hat{x}$$

$$P(C_2|x) > P(C_1|x) \text{ for } x > \hat{x}$$

# Decisions and costs

- Minimizing classification error:



# Decisions and costs

- Suppose  $C_1$  is cancer patient and  $C_2$  is healthy. It may be more costly to mistake  $C_1$  for  $C_2$  than vice-versa.
- We can consider the following **loss matrix**:

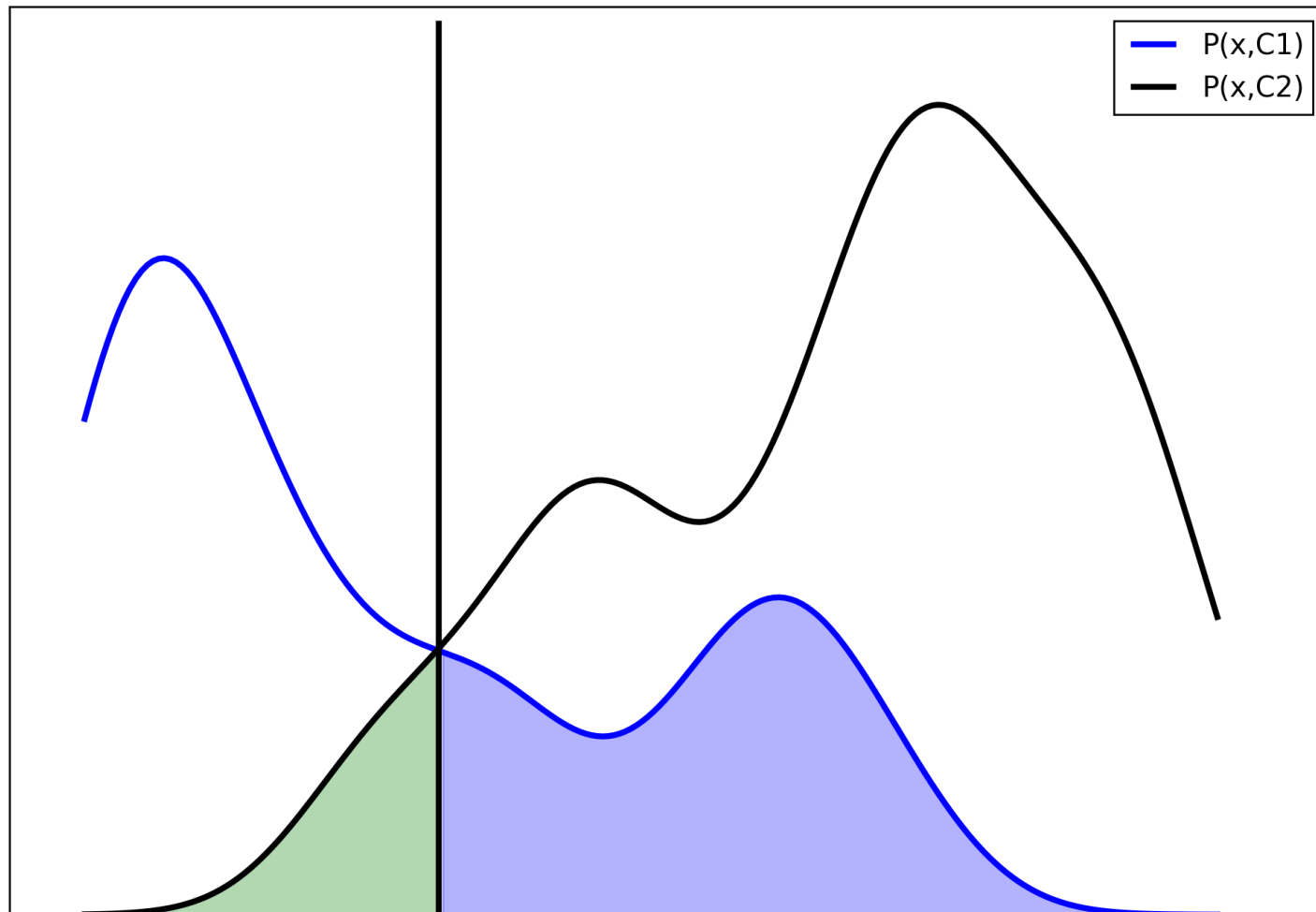
	Predict cancer	Predict healthy
Is cancer	0	5
Is healthy	1	0

- Now we classify minimizing this **loss function**:

$$\sum_k L_{k,j} p(C_k|x)$$

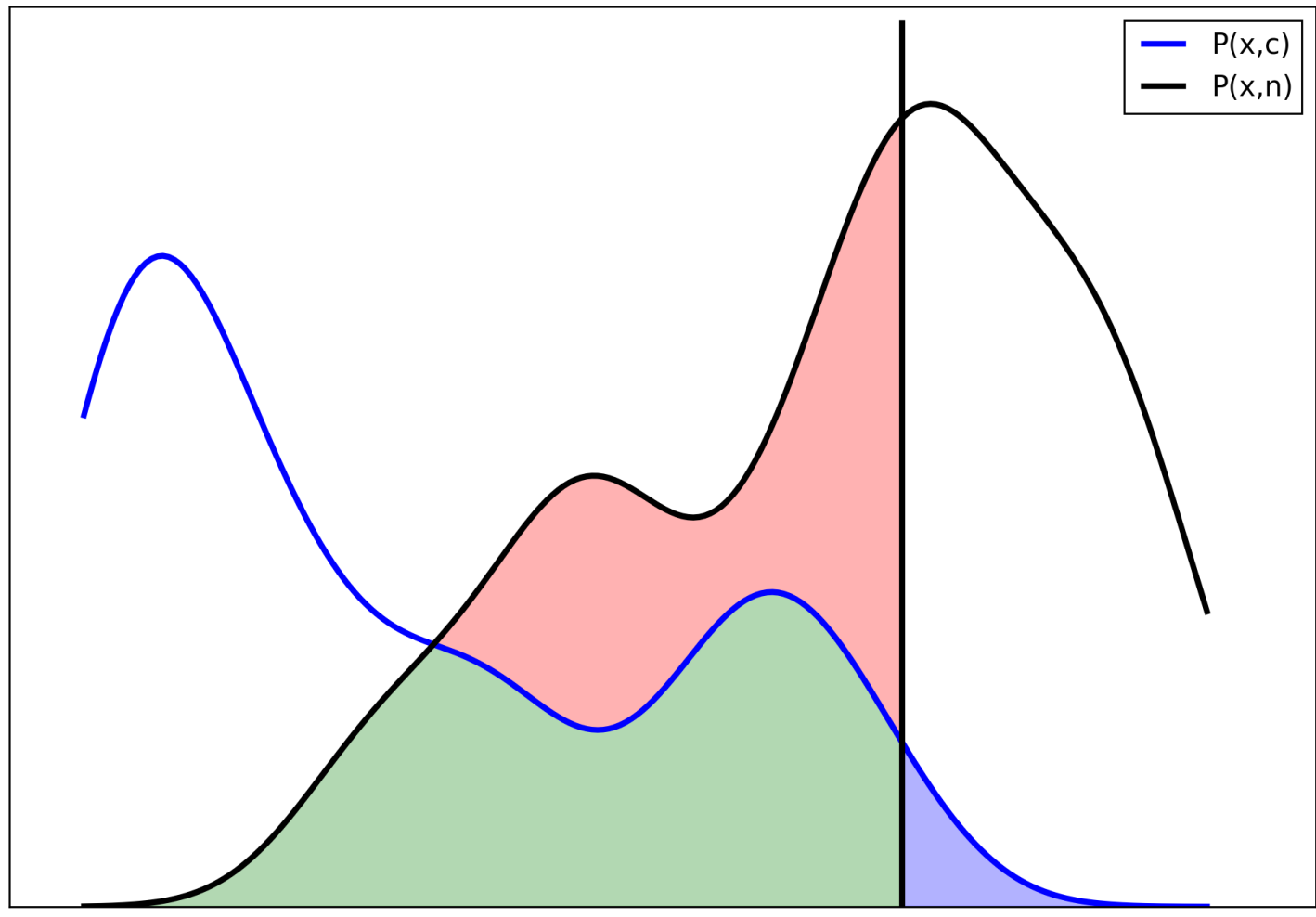
# Decisions and costs

- Minimizing classification error:



# Decisions and costs

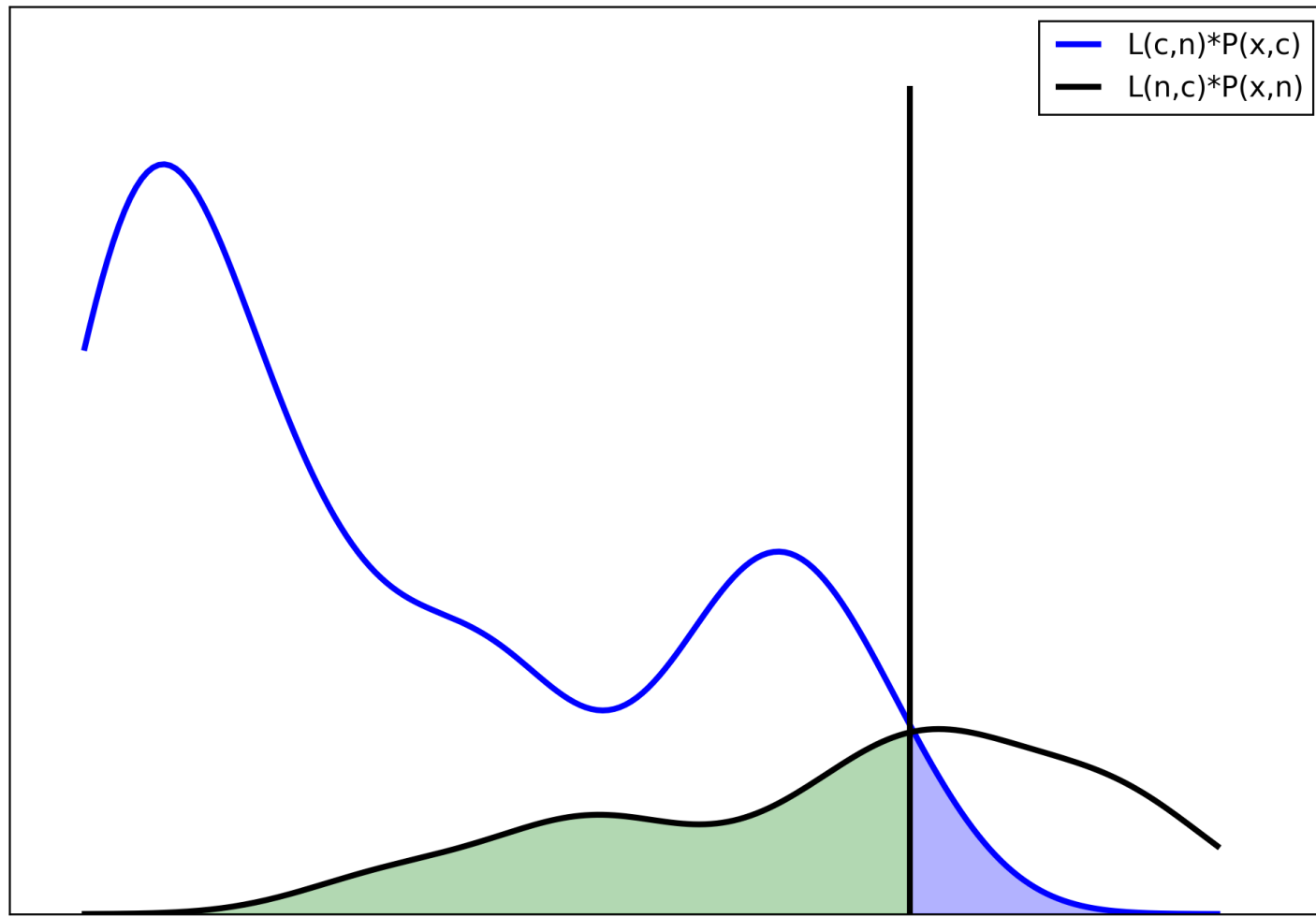
- Taking loss into account:





# Decisions and costs

- Intuition: Multiplying by misclassification cost:



# Decisions and costs

## Utility and Loss

- **Utility**: decision literature often mentions a utility function instead of a loss function
  - The idea is the same, but maximize instead of minimize

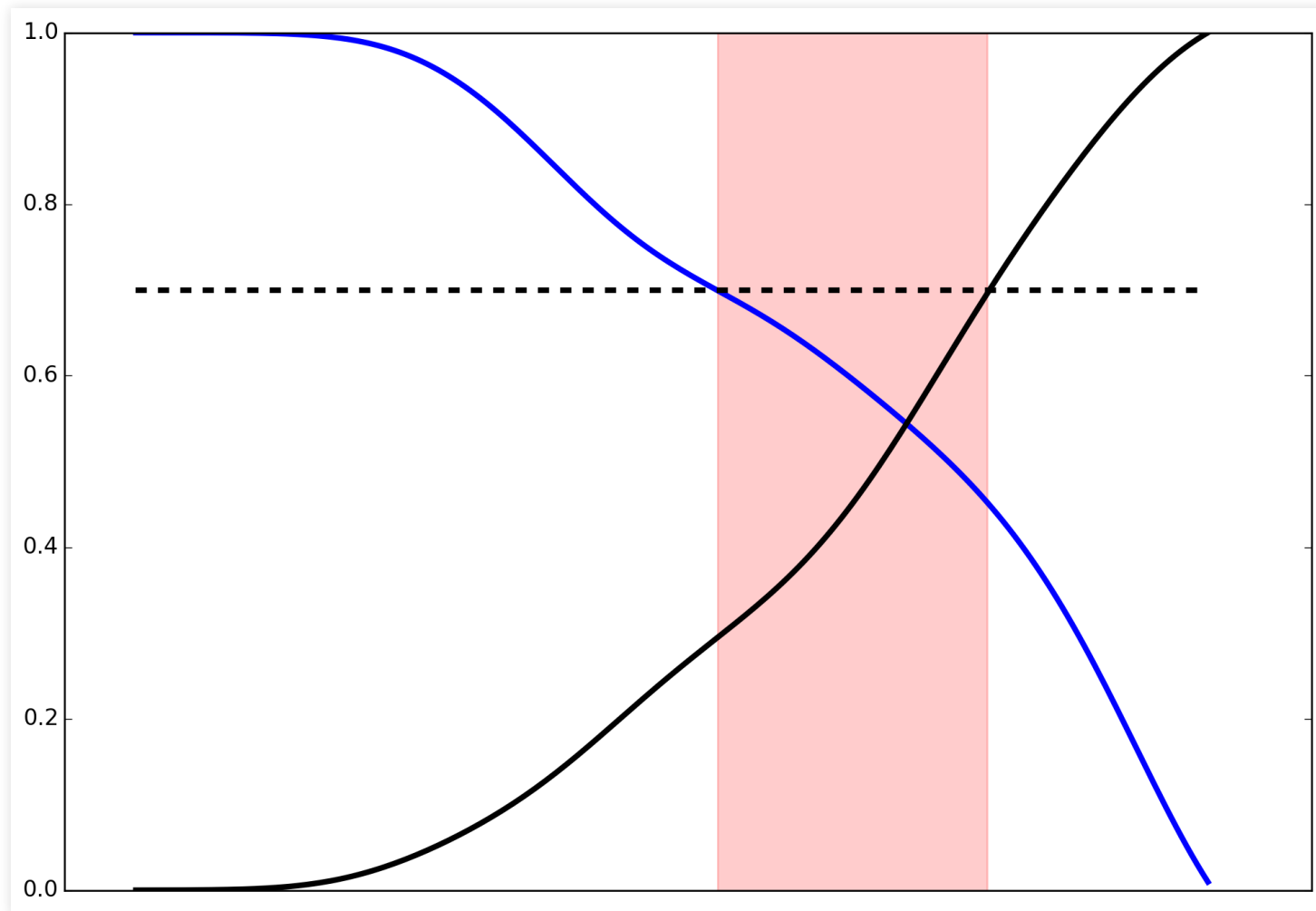
## Decision confidence

- **Rejection option**
  - Misclassification often occurs when probabilities are similar
  - We can reject classification in those cases (e.g. warn user)

$$p(C_k|x) \leq \phi \quad \forall k$$

# Decisions and costs

- Rejecting classification below 0.7



## Summary

## Summary

- Bayesian interpretation
- MAP vs ML: importancen of priors
- Decision: misclassification, cost, rejection

## Further reading

- Alpaydin, Chapter 3 up to 3.5
- Bishop, Section 1.5

