Machine Learning 1

Lecture 04 - Linear Methods for Regression - Linear Classification

Patrick Forré

Problems: Underfitting and Overfitting

- Underfitting: model not flexible/complex enough (M too low) to capture variability of true function f.

  Detection: both training and test error comparatively high.
  Possible solutions:
  - Increase parameter space $\mathcal{W}$, i.e. complexity M,
  - create additional basis functions / "features" ϕ_j of the data x,
  - measure new meaningful properties of the samples.
- Overfitting: model too flexible (M too big in comparison to number of observations N). It will start to model variance and noise instead of true underlying function.

  Detection: training error low, test error high.
  Possible solutions:
  - get more data (increase N).
  - decrease parameter space $\mathcal{W}$, i.e. lower complexity M,
  - penalize big parameters / coefficients w_i ("Shrinkage", "Weight Decay", "Regularization", "Bayesian Approach").

Linear Basis Function Model with Ridge Regularization

- Training data: $D = (x_1, \ldots, x_N)^T$ with targets $T = (t_1, \ldots, t_N)^T$, where every $x_i \in \mathbb{R}^D$ is a D-dimensional vector $x_i = (x_{i,1}, \ldots, x_{i,D})^T$.
- Fix a number M and choose basis functions/"features" of x: $(\phi_0(x), \ldots, \phi_{M-1}(x))^T =: \phi(x)$, with $\phi_0 \equiv 1$.
- Model functions with parameters $w = (w_0, \ldots, w_{M-1}) \in \mathbb{R}^M$:

$$y(x, w) = \sum_{i=0}^{M-1} w_i \cdot \phi_i(x) = w^T \phi(x).$$

- Minimize the Ridge regularized sum-of-squares error function:

$$E_{\mathrm{RG}}(D, T, w) := \frac{1}{2} \sum_{i=1}^{N} (t_i - y(x_i, w))^2 + \frac{\lambda}{2} \sum_{k=0}^{M-1} |w_k|^2.$$

- Unique minimizer: $w_{\mathrm{RG}} = (\lambda \mathbb{1}_M + \Phi^T \Phi)^{-1} \Phi^T T$, with $N \times M$-matrix Φ with entries $\Phi_{ik} = \phi_k(x_i)$.

Model Comparison and Model Selection

Question

If we have different models (e.g. different M, λ etc.) to describe the data which should we choose?

- If we have enough data then we split the data into training, validation and test data and evaluate every model (fully trained on the training set) on the validation set. Choose the one with lowest validation test error.
- If data is scarce one can use S-fold cross validation.
- One could use information criteria, which penalize complexity:
  - Akaike IC (AIC): Choose model with minimal:
$$M - \ln p(D|w_{\mathrm{ML}}).$$
  - Bayesian IC (BIC): Choose model with minimal:
$$\frac{1}{2} M \ln N - \ln p(D|w_{\mathrm{MAP}}).$$
  - Full Bayesian.

Expected Test Error: Bias - Variance - Decomposition

- Let X, ϵ be independent random variables with $\mathbb{E}[\epsilon] = 0$ and $T = h(X) + \epsilon$ and $D = (X_1, \ldots, X_N)$ i.i.d. instances of X and W a noisy parameter "learned" from D and y the predictive function. Then the expected (quadratic) test error is:

$$\begin{aligned} & \mathbb{E}[(T - y(X, W))^2] & \\ = \; & \mathbb{E}[(T - h(X))^2] & (\text{noise})^2 \\ + \; & \mathbb{E}[(h(X) - \mathbb{E}_D[y(X, W)])^2] & (\text{bias})^2 \\ + \; & \mathbb{E}[(\mathbb{E}_D[y(X, W)] - y(X, W))^2] & (\text{variance}) \end{aligned}$$

- Expected Test Error = Bias^2 + Variance + Noise^2,
- Bias: measures the "difference" between desired regression function h and the avarage prediction over all data sets.
- Variance: measures sensitivity of y to particular choice of data set around the average over all data sets.
- Noise: just a constant coming from the variance of ϵ.

Bayesian Linear Regression (I)

- Training data: $D = (x_1, \ldots, x_N)^T$ with targets $T = (t_1, \ldots, t_N)^T$.
- Linear Basis Function Model: $t = w^T \phi(x) + \epsilon$ with Gaussian noise ϵ and parameters $w = (w_0, \ldots, w_{M-1})^T \in \mathbb{R}^M$.
- So for (x, t) we have $p(t|x, w) = \mathcal{N}(t|w^T \phi(x), \beta^{-1})$, where $\beta = 1/\sigma^2$ is the precision, leading to:
- Likelihood: $p(T|w, D, \beta) = \prod_{i=1}^{N} \mathcal{N}(t_i|w^T \phi(x_i), \beta^{-1})$.
- Bayesian Approach: Gaussian Prior: $p(w) = \mathcal{N}(w|\mu_0, \Sigma_0)$ with mean μ_0 and covariance Σ_0. This leads to:
- Gaussian Posterior: $p(w|T, D, \beta) = \mathcal{N}(w|\mu_N, \Sigma_N)$ with mean μ_N and covariance Σ_N calculated to (see Matrix Cook Book):

$$\begin{aligned} \Sigma_N &= (\Sigma_0^{-1} + \beta \Phi^T \Phi)^{-1} \\ \mu_N &= \Sigma_N(\Sigma_0^{-1} \mu_0 + \beta \Phi^T T) \end{aligned}$$

- So the Maximum A Posteriori estimate is $w_{\mathrm{MAP}} = \mu_N$.

Bayesian Linear Regression (II)

- Special case: $\mu_0 = 0$ and $\Sigma_0 = \alpha^{-1}\mathbb{1}$ with $\alpha > 0$. Leading to:
- Gaussian Prior: $p(w|\alpha) = \mathcal{N}(w|0, \alpha^{-1}\mathbb{1})$.
- Gaussian Posterior: $p(w|T, D, \alpha, \beta) = \mathcal{N}(w|\mu_N, \Sigma_N)$ with:

$$\begin{aligned} \Sigma_N &= (\alpha\mathbb{1} + \beta\Phi^T\Phi)^{-1} \\ \mu_N &= \beta\Sigma_N\Phi^T T. \end{aligned}$$

- Maximizing the log-posterior (with Gaussian prior) w.r.t. w:

$$\begin{aligned} \ln p(w|T, D, \alpha, \beta) &= -\frac{\beta}{2}\sum_{i=1}^{N}(t_i - w^T\phi(x_i))^2 - \frac{\alpha}{2}w^T w + \text{const} \\ &= -\beta\left(E(D, T, w) + \frac{\alpha}{2\beta}||w||_2^2\right) + \text{const.} \end{aligned}$$

is equivalent to Ridge Regression with regularization parameter $\lambda = \frac{\alpha}{\beta}$. So
$w_{\text{MAP}} = \mu_N = w_{\text{RG}} = (\frac{\alpha}{\beta}\mathbb{1}_M + \Phi^T\Phi)^{-1}\Phi^T T$ for this choice of distributions.

Sequential Bayesian Learning for Linear Regression

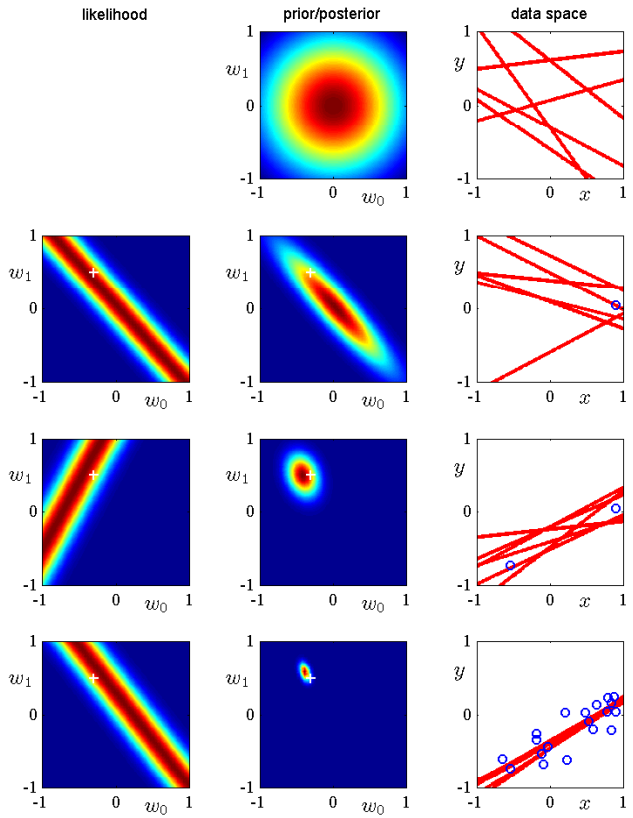

Figure: Predictive functions $y(x, w) = w_0 + w_1 \cdot x$. Likelihood $p(t|x, w)$, prior/posterior $p(w|D)$. White cross = true value (Bishop 3.7)

Bayesian Predictive Function for the Linear Model

- Training data: $D = (x_1, \ldots, x_N)^T$ with targets $T = (t_1, \ldots, t_N)^T$.
- Linear Basis Function Model: $t = w^T \phi(x) + \epsilon$ with Gaussian noise ϵ and parameters $w = (w_0, \ldots, w_{M-1})^T \in \mathbb{R}^M$.
- So $p(t|x, w, \beta) = \mathcal{N}(t|w^T \phi(x), \beta^{-1})$.
- Gaussian Posterior: $p(w|T, D, \beta) = \mathcal{N}(w|\mu_N, \Sigma_N)$ with mean μ_N and covariance Σ_N was:

$$\begin{aligned} \Sigma_N &= (\Sigma_0^{-1} + \beta \Phi^T \Phi)^{-1} \\ \mu_N &= \Sigma_N(\Sigma_0^{-1}\mu_0 + \beta \Phi^T T) \end{aligned}$$

- The predictive distribution then is:

$$\begin{aligned} p(t|x, T, D, \beta) &= \int p(t|x, w, \beta) p(w|T, D, \beta) dw \\ &= \mathcal{N}(t|\mu_N^T \phi(x), \sigma_N), \end{aligned}$$

with $\sigma_N = \frac{1}{\beta} + \phi(x)^T \Sigma_N \phi(x)$, which is

- a sum of noise term and a term which goes to zero for $N \to \infty$, reflecting the uncertainty of w.

Example: Bayesian Predictive Distributions

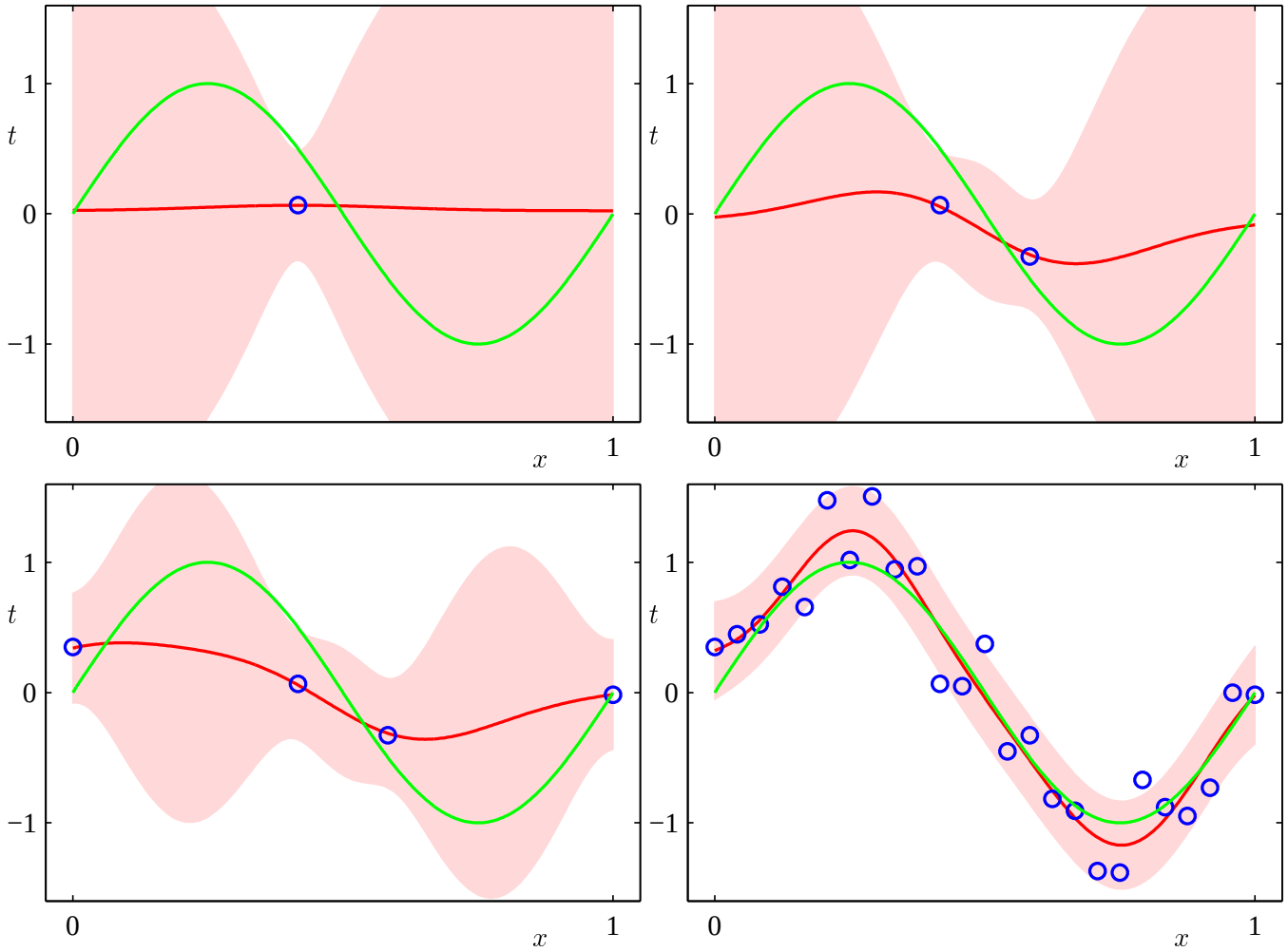

Figure: Predictive distributions for $h(x) = \sin(2\pi x)$ (green) plus noise. 9 Gaussian basis functions. Number of observations: $N = 1, 2, 4, 25$. Red—mean distribution plus one standard deviation. (Bishop 3.8)

Example: Samples from Bayesian Predictive Distributions

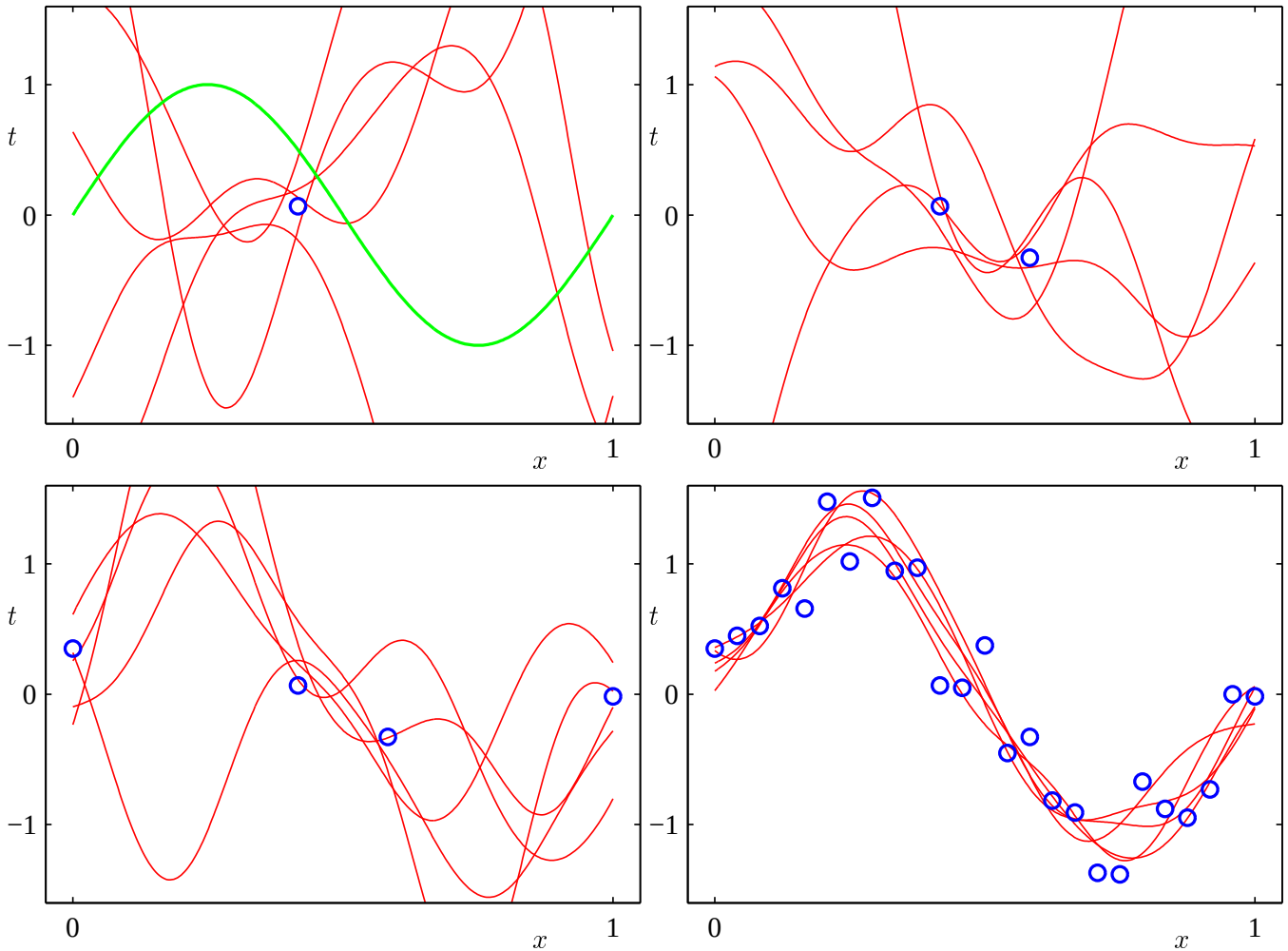

Figure: Sample functions $y(x, w)$ drawn from Bayesian predictive distributions (Bishop 3.9)

Bayesian Model Comparison (I)

- Given models $\mathcal{M}_i$, $i = 1, \ldots, L$, consisting of probability distributions and each having its own set of parameters $\mathcal{W}_i$.
- Bayesian approach: Uncertainty is expressed by a prior probability distributions over the set of models: $p(\mathcal{M}_i)$.
- After observing data D we can reevaluate the uncertainty by computing the posterior distribution:

$$p(\mathcal{M}_i|D) = \frac{p(D|\mathcal{M}_i)}{p(D)} p(\mathcal{M}_i).$$

- As predictive function then one can use the model average:

$$p(t|x, D) = \sum_{i=1}^{L} p(t|x, D, \mathcal{M}_i) p(\mathcal{M}_i|D),$$

- or the maximal a posterior distribution $p(t|x, D, \mathcal{M}_i)$, where $i = \text{argmax}_k \, p(\mathcal{M}_k|D)$.

Bayesian Model Comparison (II)

- For the last approach we need to compare $\mathcal{M}_i$ and $\mathcal{M}_j$, i.e. we need to evaluate the quotient:

$$\frac{p(\mathcal{M}_i|D)}{p(\mathcal{M}_j|D)} = \frac{p(D|\mathcal{M}_i)}{p(D|\mathcal{M}_j)} \frac{p(\mathcal{M}_i)}{p(\mathcal{M}_j)}.$$

- If either the quotient of priors $\frac{p(\mathcal{M}_i)}{p(\mathcal{M}_j)}$ is given or is close to 1 (e.g. if $p(\mathcal{M}_k) = \frac{1}{L}$) then we are left to evaluating the Bayes factor:

$$K_{ij} := \frac{p(D|\mathcal{M}_i)}{p(D|\mathcal{M}_j)}.$$

- The marginal likelihood $p(D|\mathcal{M}_i)$ will also be called model evidence and plays the central role in Bayesian model comparison. Caution: don't confuse with $p(D)$. We have:

$$p(D|\mathcal{M}_i) = \int_{\mathcal{W}_i} p(D|w, \mathcal{M}_i) p(w|\mathcal{M}_i) dw.$$

Bayesian Model Comparison for Linear Basis Function Model

- Linear Basis Function Models:

  $$\mathcal{M}_M = (M; \mathcal{W}_M = \mathbb{R}^M; \phi_0, \ldots, \phi_{M-1}; y(x, w) = w^T \phi(x); \alpha, \beta)$$

- Training data: $D = (x_1, \ldots, x_N)^T$ with targets $T = (t_1, \ldots, t_N)^T$.
- Likelihood: $p(T|w, D, \beta, M) = \prod_{i=1}^{N} \mathcal{N}(t_i|w^T \phi(x_i), \beta^{-1})$.
- Prior: $p(w|\alpha, M) = \mathcal{N}(w|0, \alpha^{-1} \mathbb{1}_M)$.
- Posterior: $p(w|T, D, \alpha, \beta, M) = \mathcal{N}(w|\mu_N, \Sigma_N)$ with:

  $$\begin{aligned} \Sigma_N &= (\alpha \mathbb{1}_M + \beta \Phi^T \Phi)^{-1} \\ \mu_N &= \beta \Sigma_N \Phi^T T. \end{aligned}$$

- We get the log Model Evidence (by Bayes' rule):

  $$\ln p(T|D, \alpha, \beta, M) = \frac{M}{2} \ln \alpha + \frac{N}{2} \ln \beta - E_{\mathrm{RG}}(\mu_N) + \frac{1}{2} \ln |\Sigma_N| - \ln(2\pi)$$

  with $E_{\mathrm{RG}}(\mu_N) = \frac{\beta}{2} ||T - \Phi \mu_N||_2^2 + \frac{\alpha}{2} ||\mu_N||_2^2$.
- Model Selection: Choose the one with highest model evidence.

Example: Bayesian Model Comparison: Polynomials

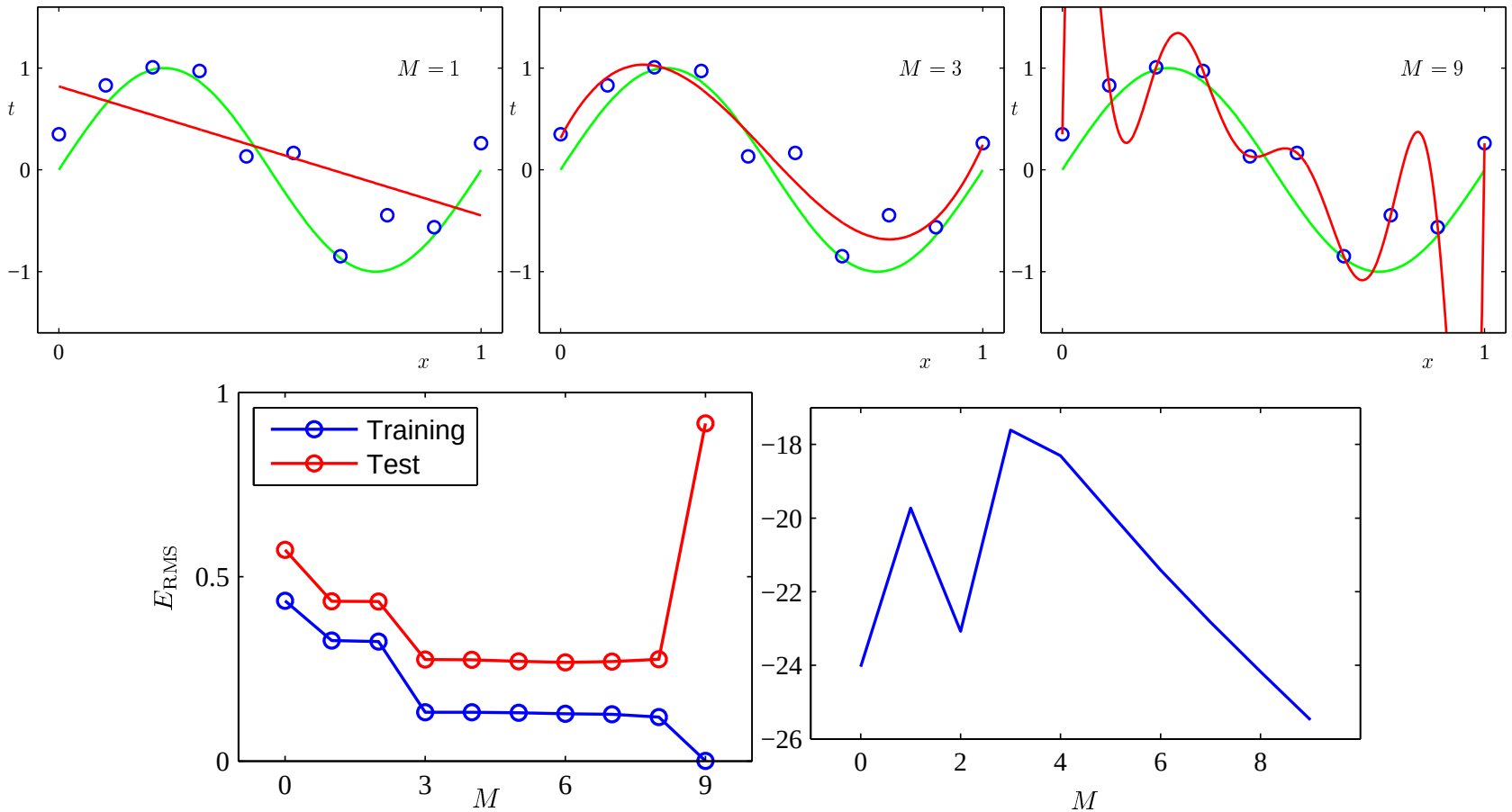

Figure: Bottom right: Model evidence by polynomial order M for polynomial regression for sinoidal function, α, β fixed. Best model with highest model evidence: $M = 3$. (Bishop 3.14)

Limitations of the Linear Methods for Regression

- Basis functions need to be given or handcrafted (not learned from data).
- Curse of dimension: To cover growing dimensions D of input vectors the number of basis functions need to grow rapidly, often exponentially.

Linear Methods - Further Reading

- Subset Selection (selecting the most important features ϕ_i out of the given ones).
- Variance Analysis (ANOVA) of the estimators.
- Testing for zero coefficients.
- Analysis of the residual distribution (e.g. testing for normality).
- Outlier analysis.
- Other regularization techniques.

Supervised Learning: Classification

- Given an input vector $x = (x_{,1}, \ldots, x_{,D}) \in \mathbb{R}^D$ we want to assign it to / predict one of the K classes $t \in \{c_1, \ldots, c_K\}$.
- The strategy will be to devide $\mathbb{R}^D$ into decision regions each assigned to a class and whose boundaries are called decision bounderies.

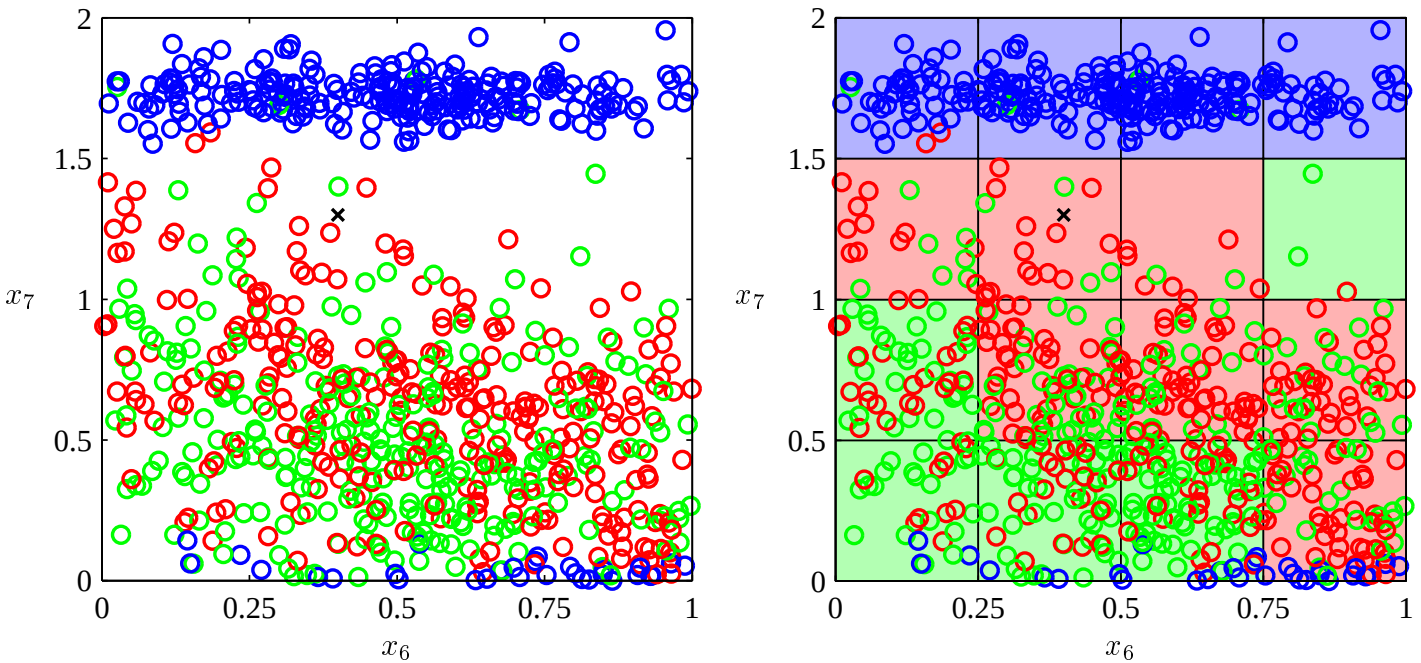

Figure: Classification via decision regions (Bishop 1.19 + 1.20)

Linear Classification

- Linear classification means that we consider linear $(D-1)$-dimensional hyperplanes as decision boundaries.
- Data sets whose classes can be separated exactly by linear decision surfaces are called linear separable.

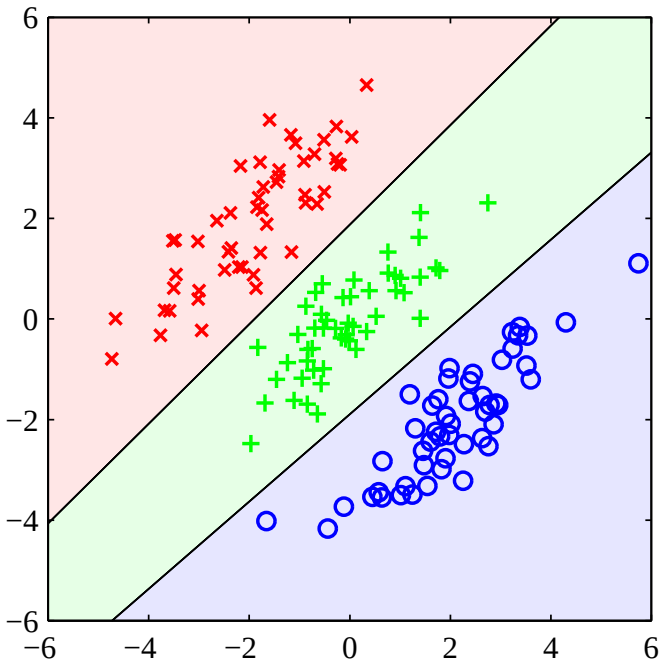

Figure: Linear separable data set (Bishop 4.5)

Multiple Classes: one-vs-the-rest dummies

- Situation: Predict one of the K classes $\{c_1, \ldots, c_K\}$ of a random variable T with $K \geqslant 2$.
- For $j = 1, \ldots, K$ define the one-vs-the-rest dummy variable:

$$\mathbb{1}_{c_j}(T) := \begin{cases} 1 & \text{if} \quad T = c_j \\ 0 & \text{if} \quad T \neq c_j. \end{cases}$$

- l.a.w. represent c_j as the vector $(0, \ldots, 0, \overbrace{1}^{j\text{-th}}, 0, \ldots, 0)^T$.
- Predicting the K-classed variable $T \in \{c_1, \ldots, c_K\}$ is then equivalent to the K-fold binary prediction of $\mathbb{1}_{c_j}(T) \in \{0, 1\}$ for $j = 1, \ldots, K$.
- So in most cases we can reduce to the case where T is a binary variable with classes $\{0, 1\}$. But not always:

Example: one-vs-the-rest failure

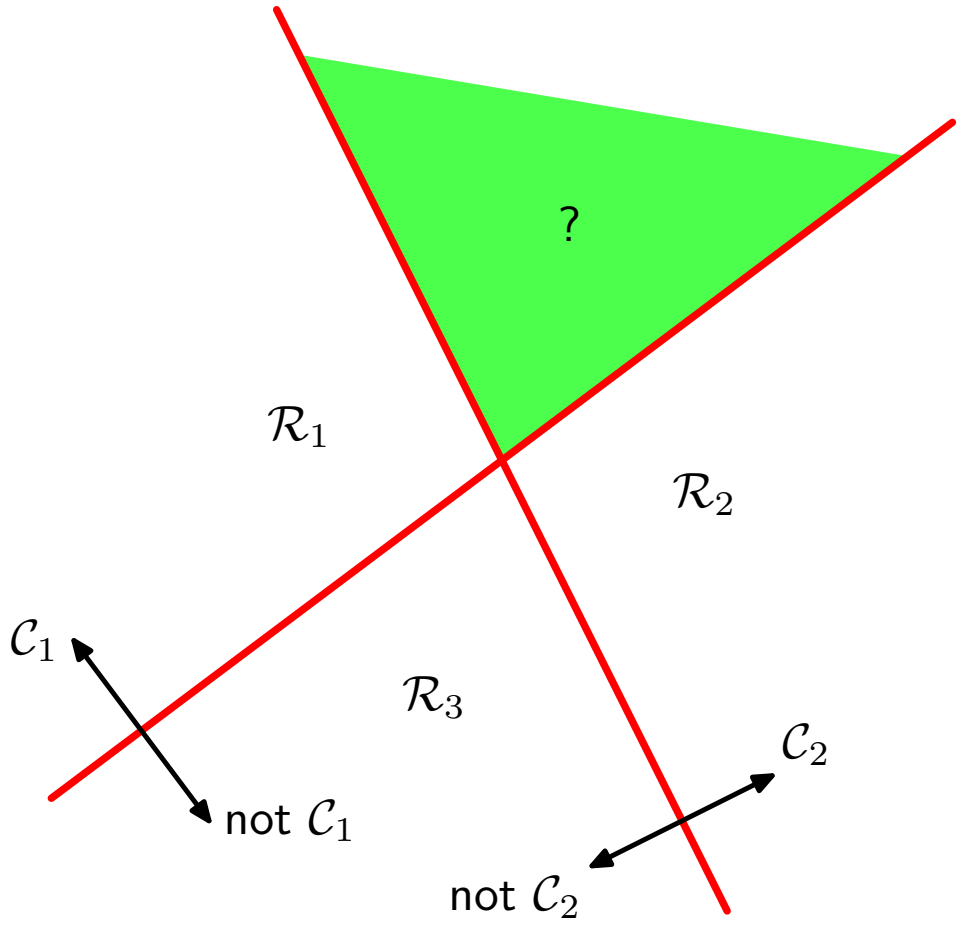

Figure: The one-vs-the-rest construction for $K \geqslant 3$ classes leading to ambiguous regions (green) (Bishop 4.2)

Classification: Three approaches

We will analyse three different approaches for the classification task:

1. Discriminant Functions: Learn a function $y(x, w)$ assigning x into $\{c_1, \ldots, c_K\}$.
   We will consider generalized linear discriminant functions of the form:
   $$y(x, w) = g(\sum_{m=0}^{M} w_m \phi_m(x)),$$
   where ϕ_m are "features" of x and g is a (non-linear) activation function. For simplicity we will assume $\phi_m(x) = x_m$.
2. Probabilistic Generative Models: Model the class-conditional densities $p(x|c_j)$ as well as the class priors $p(c_j)$, and then use Bayes' rule to compute the posterior density $p(c_j|x)$.
3. Probabilistic Discriminative Models: Maximize a likelihood function attached to the density $p(c_j|x)$.

Linear Discriminant Functions: Two Classes

- For D-dimensional input vector $x = (x_{,1}, \ldots, x_{,D})^T \in \mathbb{R}^D$ and two classes $\{c_0, c_1\}$, in the simplest case, we consider real valued linear linear discriminant functions:

$$y(x, w) = w^T x + w_0,$$

  where $w \in \mathbb{R}^D$ is called weight vector and $w_0 \in \mathbb{R}$ the bias.
- $\mathcal{B} = \{x \in \mathbb{R}^D | y(x, w) = 0\}$ is called the decision boundary.
- We then have the decision regions for x given by
  $\mathcal{R}_0 = \{x \in \mathbb{R}^D | y(x, w) < 0\}$ (for class c_0) and
  $\mathcal{R}_1 = \{x \in \mathbb{R}^D | y(x, w) > 0\}$ (for class c_1).
- The vector w stands orthogonal onto the decision boundary and points into the c_1-region:
  If $y(x_A, w) = 0$ and $y(x_B, w) = 0$ then $w^T(x_A - x_B) = 0$.
  If $y(x_C, w) > 0$ then $w^T(x_C - x_A) > 0$.
- w_0 determines the signed normal distance of the decision boundary from the origin $x_O = 0$: $\frac{w^T x_A}{||w||} = -\frac{w_0}{||w||}$.

Example: Geometry of Linear Discriminant Functions

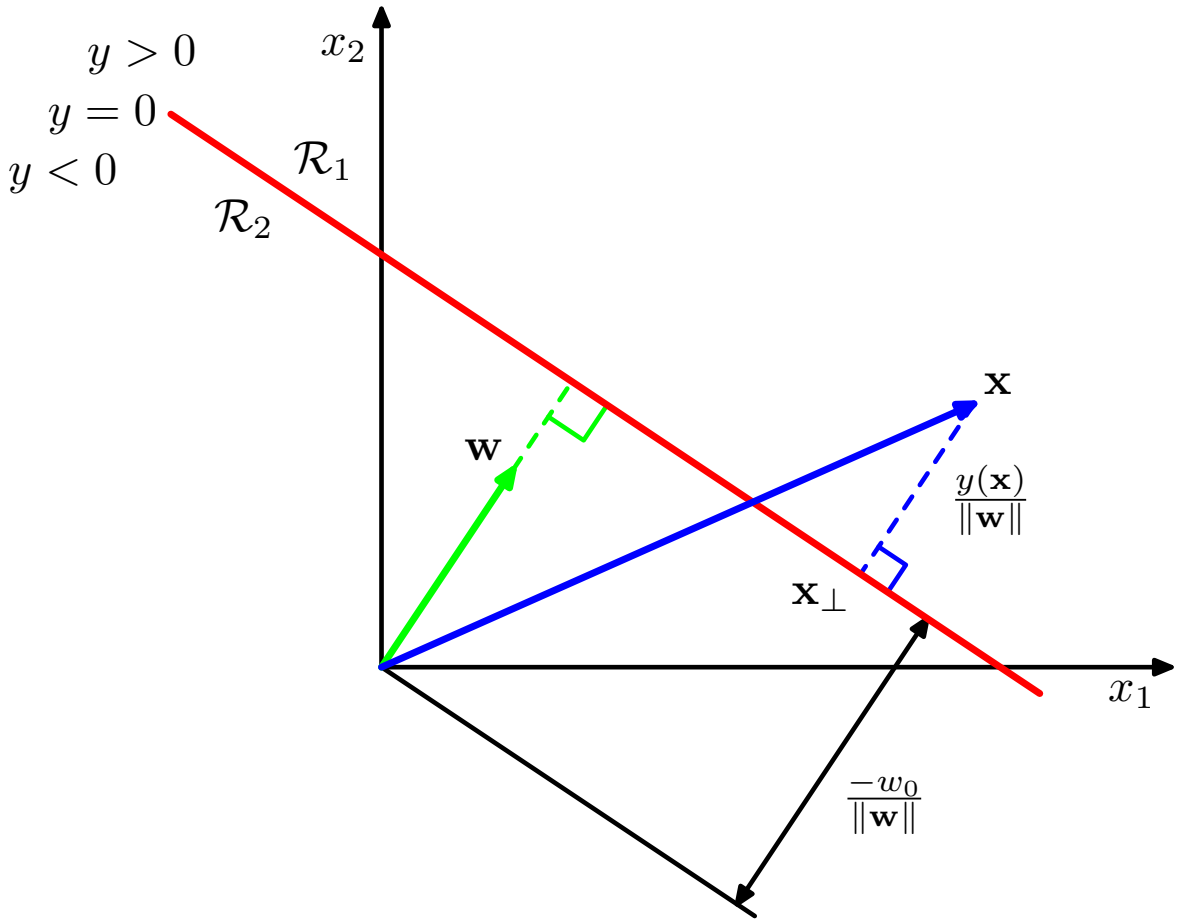

Figure: Decision surface $y(x) = 0$ in red is orthogonal to w. Signed normal distance of a point x to the decision surface is $y(x)/||w||$ in blue. (Bishop 4.1)

Linear Discriminant Functions: Multiple Classes

- For D-dimensional input vector $x = (x_{,1}, \ldots, x_{,D})^T \in \mathbb{R}^D$ and K classes $\{c_1, \ldots, c_K\}$, we now consider the K linear functions:

$$y_k(x) = w_k^T x + w_{k,0},$$

  where every $w_k \in \mathbb{R}^D$ and $k = 1, \ldots, K$.
- The region for assigning an x to class c_k then is:

$$\mathcal{R}_k = \{x \in \mathbb{R}^D | y_k(x) > y_j(x) \forall j \neq k\}.$$

- The decision boundary $\mathcal{B}_{kj}$ between c_k and c_j is given by the $(D-1)$-dimensional hyperplane:

$$\begin{aligned} \mathcal{B}_{kj} &= \{x \in \mathbb{R}^D | y_k(x) = y_j(x)\} \\ &= \{x \in \mathbb{R}^D | (w_k - w_j)^T x + (w_{k,0} - w_{j,0}) = 0\}. \end{aligned}$$

- The regions $\mathcal{R}_k$ are convex and connected.

Example: Linear Discriminant Functions for Multiple Classes

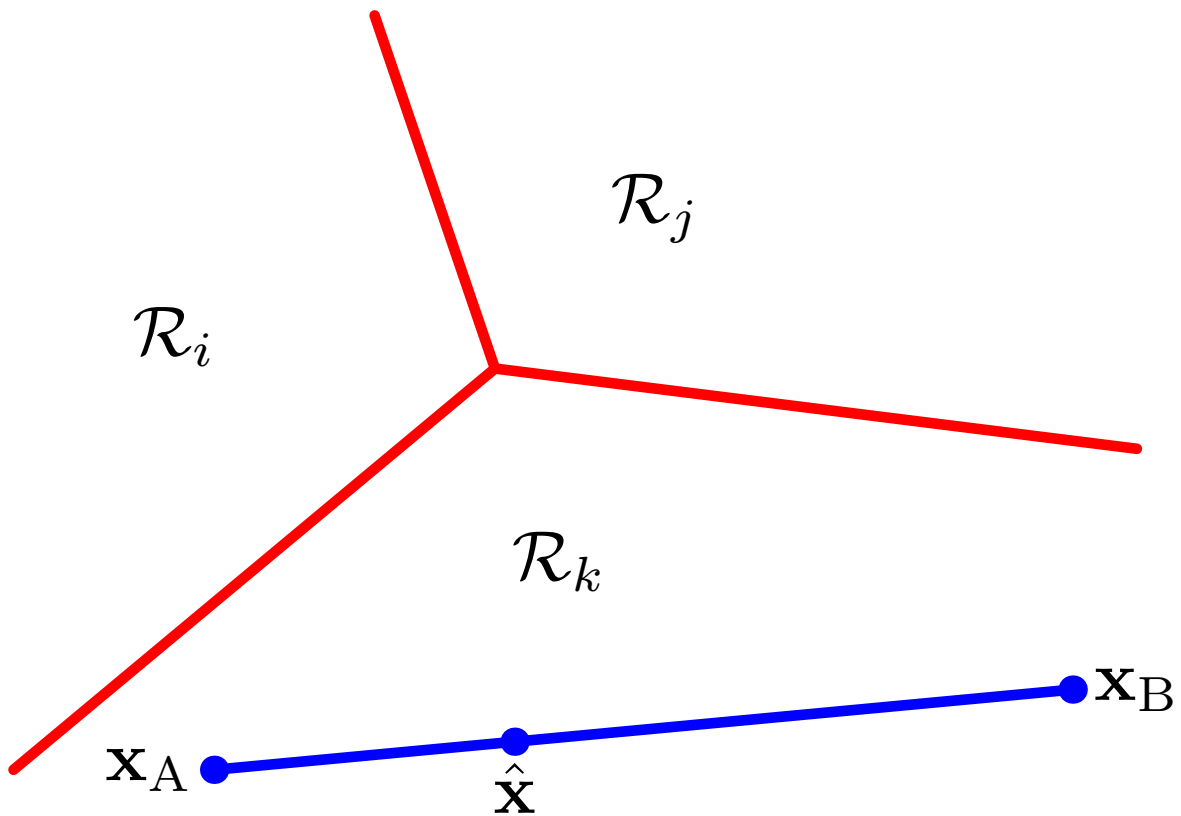

Figure: Decision regions for multiclass linear discriminant. Decision boundaries in red. The blue line illustrates the convexity and connectedness of the decision regions. (Bishop 4.3)