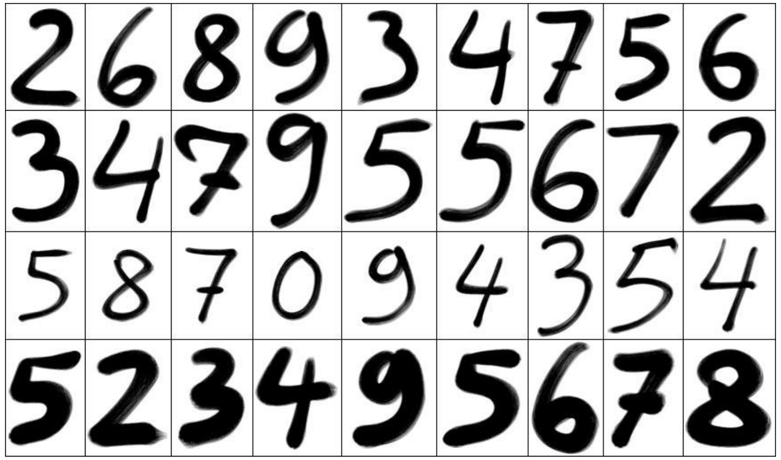
《机器学习基础》

机器学习概述

https://github.com/styluck/mech_learn

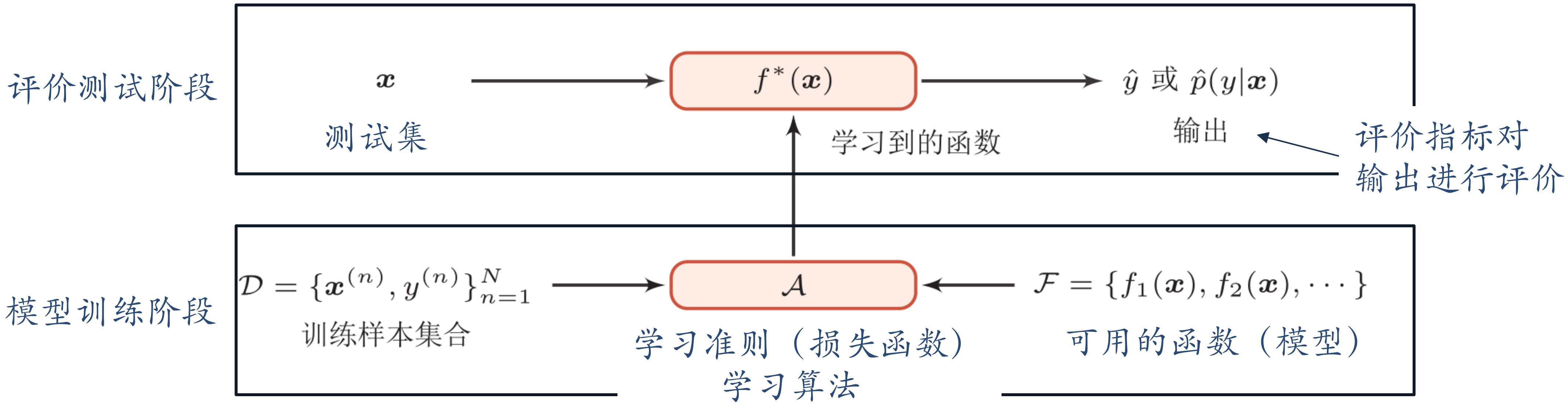
为什么要“机器学习”？

- **现实世界的问题都比较复杂**
  - 很难通过规则来手工实现



机器学习概览

- 机器学习：通过算法使得机器能从大量数据中学习规律从而对新的样本做决策。



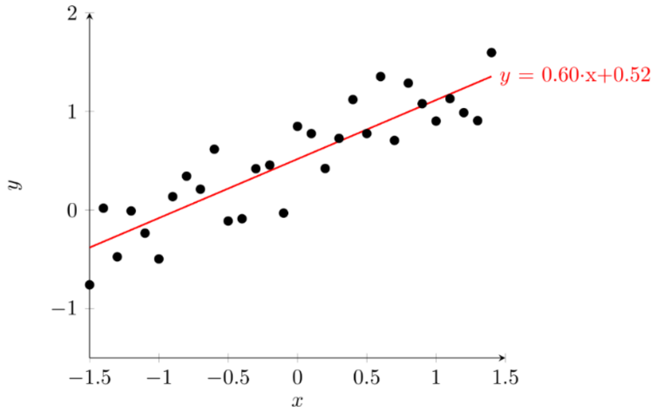
机器学习的三要素

- **模型**
  - 线性方法：$f(\mathbf{x},\theta)=\mathbf{w}^{\mathrm{T}}\mathbf{x}+b$
  - 广义线性方法：$f(\mathbf{x},\theta)=\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})+b$
    - 如果$\phi(x)$为可学习的非线性基函数，$f(\mathbf{x},\theta)$就等价于神经网络。
- **学习准则**
  - 期望风险

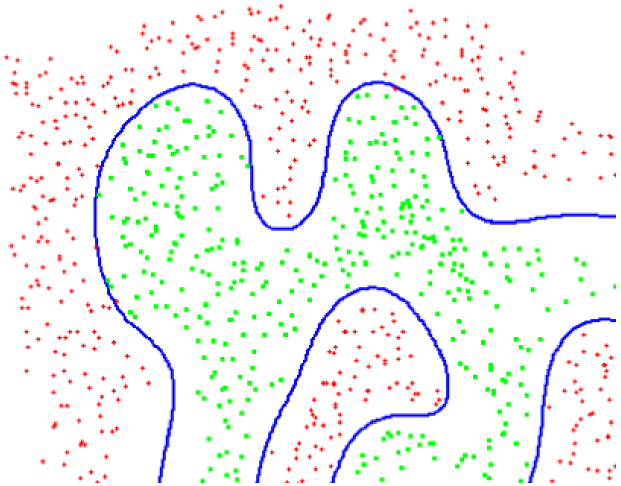
$$\mathcal{R}(f)=\mathbb{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}[\mathcal{L}(f(\mathbf{x}),y)],$$

- **优化**
  - 梯度下降

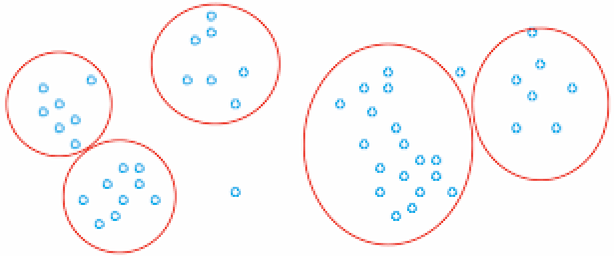
常见的机器学习问题



回归



分类

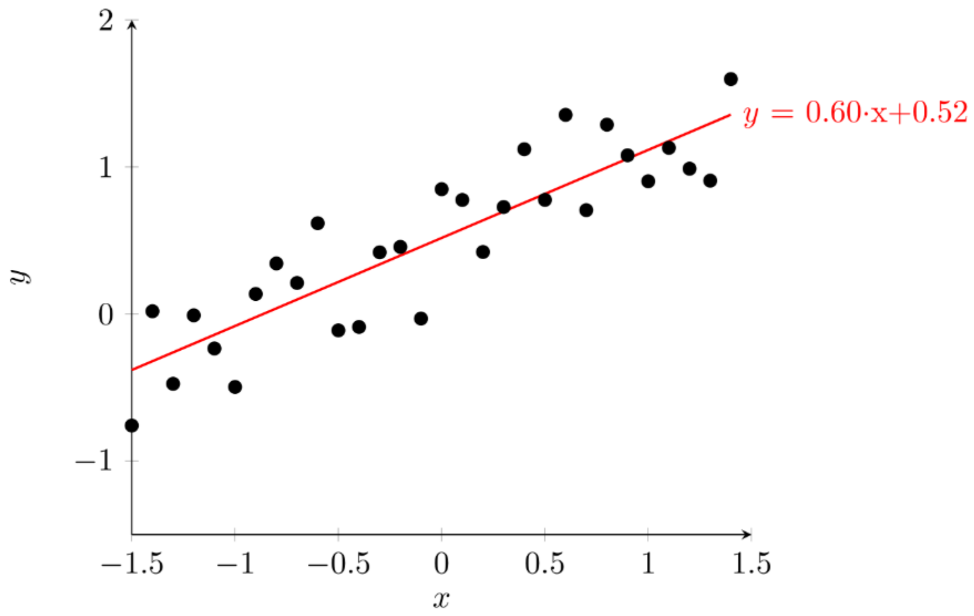


聚类

模型

- 线性模型的假设空间为一个参数化的线性函数族，即

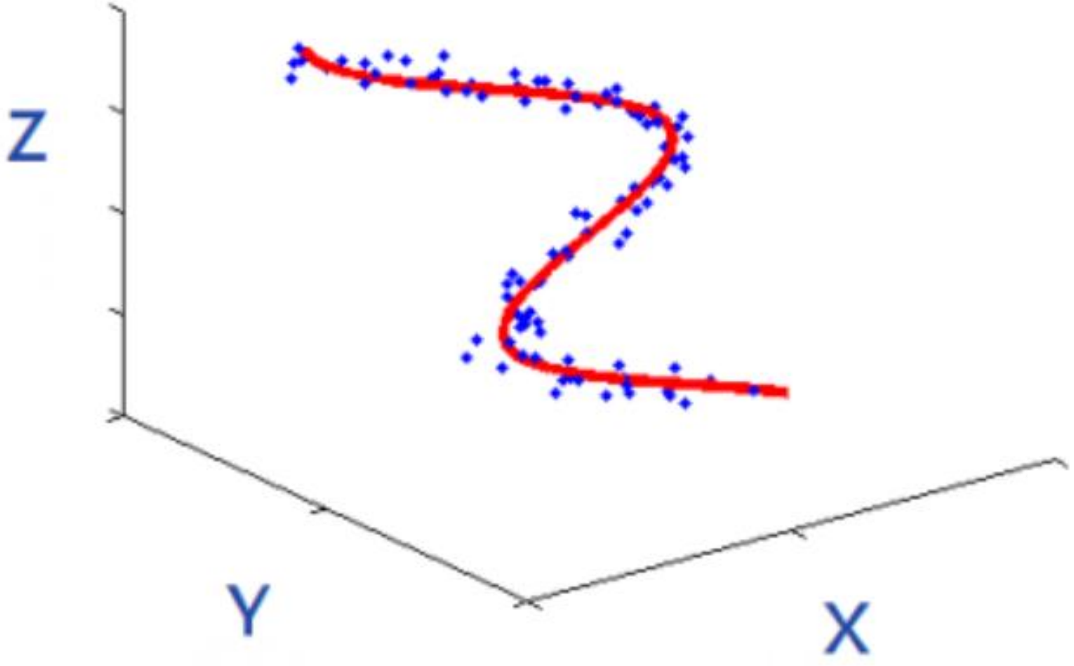
$$f(\mathbf{x};\mathbf{w},b)=\mathbf{w}^{\mathrm{T}}\mathbf{x}+b$$



模型

- 非线性模型可以写为多个非线性基函数$\Phi(x)$的线性组合

$$f(\boldsymbol{x};\theta)=\boldsymbol{w}^{\top}\phi(\boldsymbol{x})+b,$$



学习准则

- 训练数据：$\mathcal{D}=\{\mathrm{x}^{(\mathrm{n})},y^{(\mathrm{n})}\},i\in[1,N]$，由 N 个独立同分布的（Independent and Identically Distributed， IID） 样本组成，即每个样本$(\boldsymbol{x},y)\in\mathcal{X}\times\mathcal{Y}$ 是从$\mathcal{X}$ 和$\mathcal{Y}$的联合空间中按照某个未知分布$p_r(\boldsymbol{x},y)$独立地随机产生

- 一个好的模型 $f(\boldsymbol{x},\theta^*)$ 应该在所有 $(\boldsymbol{x},y)$ 的可能取值上都与真实映射函数$y=g(\boldsymbol{x})$一致， 即

$$|f(\boldsymbol{x},\theta^*)-y|<\epsilon,\qquad \forall(\boldsymbol{x},y)\in\mathcal{X}\times\mathcal{Y},$$

- 或与真实条件概率分布$pr(y|\boldsymbol{x})$一致， 即

$$|f_y(\boldsymbol{x},\theta^*)-p_r(y|\boldsymbol{x})|<\epsilon,\qquad \forall(\boldsymbol{x},y)\in\mathcal{X}\times\mathcal{Y},$$

学习准则

- 模型$f(\boldsymbol{x};\theta)$ 的好坏可以通过期望风险（Expected Risk）$\mathcal{R}(\theta)$来衡量，其定义为

$$\mathcal{R}(\theta)=\mathbb{E}_{(\boldsymbol{x},y)\sim p_r(\boldsymbol{x},y)}\left[\mathcal{L}\big(y,f(\boldsymbol{x};\theta)\big)\right],$$

- 其中$p_r(\boldsymbol{x},y)$为真实的数据分布，$\mathcal{L}(y,f(\boldsymbol{x};\theta))$为损失函数，用来量化两个变量之间的差异

损失函数

- 0-1损失函数：最直观的损失函数，但其缺点是数学性质不是很好

$$\mathcal{L}(y, f(\boldsymbol{x};\theta)) = \begin{cases} 0 & \text{if } y = f(\boldsymbol{x};\theta) \\ 1 & \text{if } y \neq f(\boldsymbol{x};\theta) \end{cases}$$

- 平方损失函数（Mean Squared Error，MSE）：计算预测值与真实值之间的平方差，常用于回归问题。

$$L(y, f(x,\theta)) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i,\theta))^2$$

损失函数

- Mean Absolute Error (MAE): 与 MSE 类似，但对异常值的鲁棒性更强。

$$L(y, f(x,\theta)) = \frac{1}{n}\sum_{i=1}^{n}|y_i - f(\boldsymbol{x}_i,\theta)|$$

- 交叉熵损失函数（Cross-Entropy Loss）：常用于分类任务，能很好地反映模型预测的概率分布与真实分布之间的差异。

$$L(y, f(x,\theta)) = -\sum_{i=1}^{n} y_i \log f(\boldsymbol{x}_i,\theta)$$

风险最小化准则

- 期望风险未知，通过经验风险近似
  - 训练数据：$\mathcal{D} = \{\mathrm{x}^{(\mathrm{n})}, y^{(\mathrm{n})}\}, i \in [1, N]$

$$\mathcal{R}_{\mathcal{D}}^{emp}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(y^{(n)}, f(x^{(n)}, \theta))$$

- 经验风险最小化
  - 在选择合适的风险函数后，我们寻找一个参数θ^*，使得经验风险函数最小化。

$$\theta^* = \arg\min_{\theta} \mathcal{R}_{\mathcal{D}}^{emp}(\theta)$$

- 机器学习问题转化成为一个最优化问题

如何选择一个合适的模型?

- 模型选择
  - 拟合能力强的模型一般复杂度会比较高，容易过拟合。
  - 如果限制模型复杂度，降低拟合能力，可能会欠拟合。
- 偏差与方差分解
  - 期望错误可以分解为

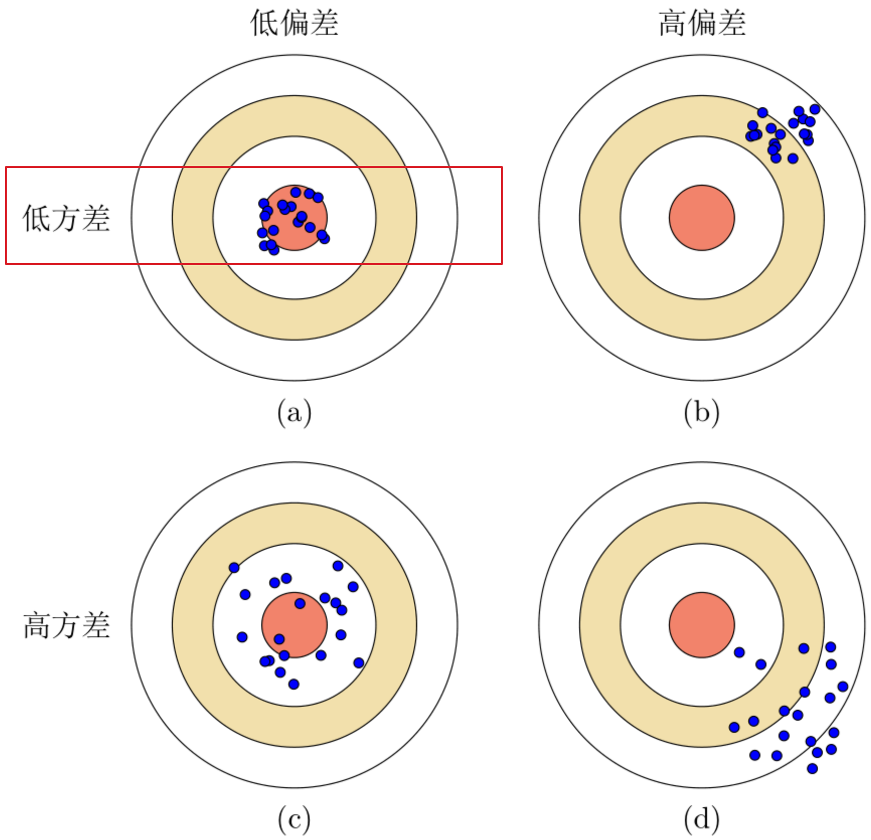
$$\mathcal{R}(f) = (\text{bias})^2 + \text{variance} + \varepsilon.$$

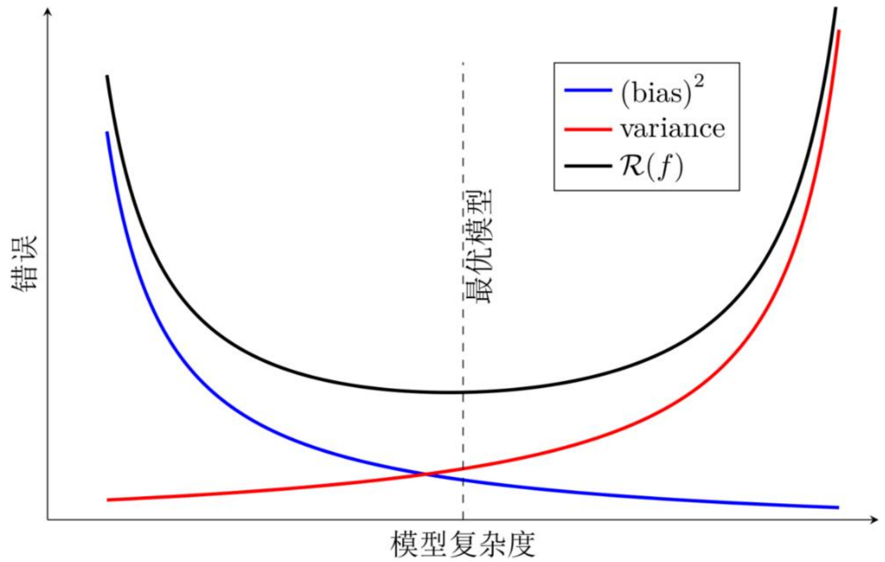
$$\mathbb{E}_{\mathbf{x}}\left[\left(\mathbb{E}_{\mathcal{D}}\left[f_{\mathcal{D}}(\mathbf{x})\right] - f^*(\mathbf{x})\right)^2\right]$$

$$\mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(f_{\mathcal{D}}(\mathbf{x}) - \mathbb{E}_{\mathcal{D}}\left[f_{\mathcal{D}}(\mathbf{x})\right]\right)^2\right]\right]$$

$$\mathbb{E}_{(\mathbf{x},y)\sim p_r(\mathbf{x},y)}\left[\left(y - f^*(\mathbf{x})\right)^2\right]$$

模型选择：偏差与方差





集成模型：有效的降低方差的方法

- 集成模型

$$f^{(c)}(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M} f_m(\mathbf{x})$$

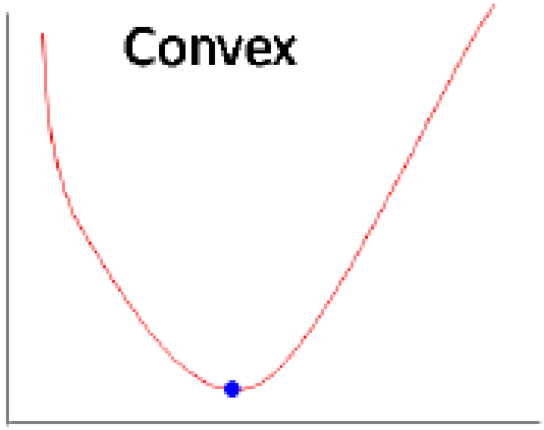
  - 通过多个高方差模型的平均来降低方差。

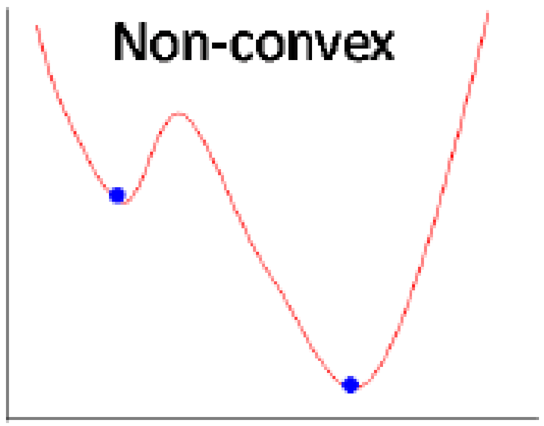
- 集成模型的期望错误大于等于所有模型的平均期望错误的1/M，小于等于所有模型的平均期望错误。

$$\bar{\mathcal{R}}(f) \geq \mathcal{R}(f^{(c)}) \geq \frac{1}{M}\bar{\mathcal{R}}(f)$$

最优化问题

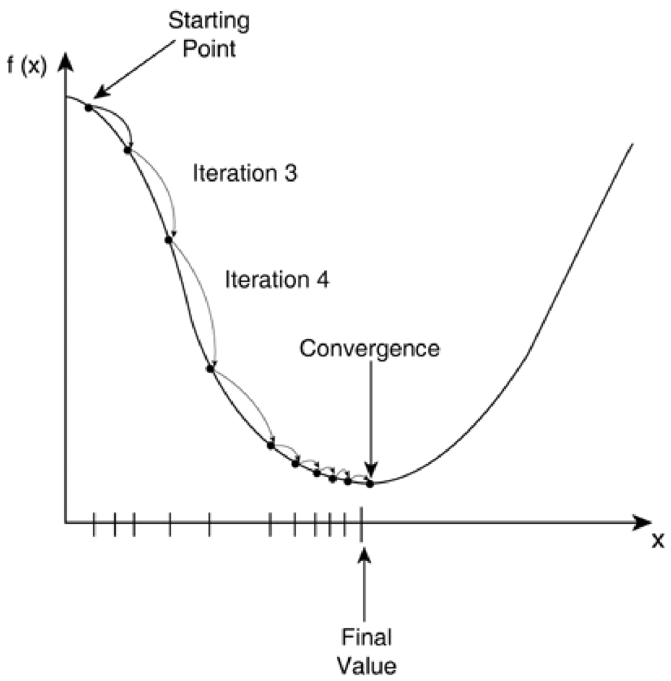
- 机器学习问题转化成为一个最优化问题

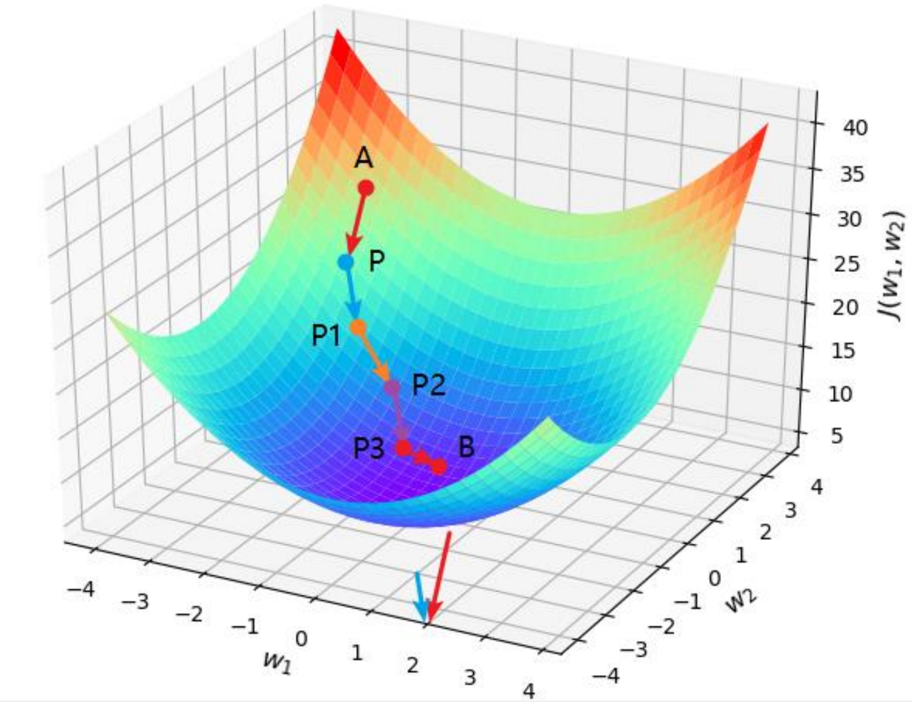




$$\min_{\theta} f(\boldsymbol{x}, \theta)$$

梯度下降法（Gradient Descent）





搜索步长α中也叫作学习率（Learning Rate）

$$
\begin{aligned}
\theta_{t+1} &= \theta_t - \alpha \frac{\partial \mathcal{R}(\theta)}{\partial \theta_t} \\
&= \theta_t - \alpha \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \mathcal{L}\left(\theta_t; x^{(i)}, y^{(i)}\right)}{\partial \theta}.
\end{aligned}
$$

随机梯度下降法

算法 2.1: 随机梯度下降法

输入: 训练集 $\mathcal{D}=\{(\mathbf{x}^{(n)},y^{(n)})\}_{n=1}^{N}$，验证集 $\mathcal{V}$，学习率 α

随机初始化 θ;

repeat

 对训练集 $\mathcal{D}$ 中的样本随机重排序;

 for $n=1\cdots N$ **do**

  从训练集 $\mathcal{D}$ 中选取样本 $(\mathbf{x}^{(n)},y^{(n)})$;

  // 更新参数

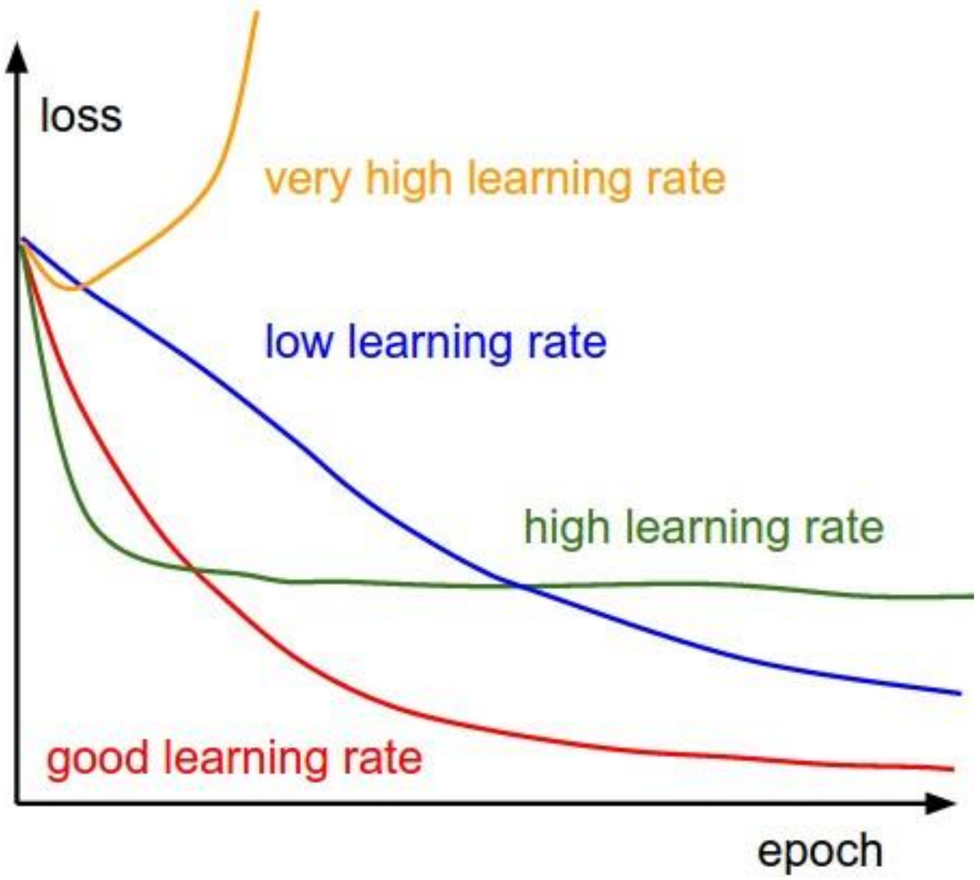
  $$\theta \leftarrow \theta - \alpha\frac{\partial\mathcal{L}(\theta;x^{(n)},y^{(n)})}{\partial\theta};$$

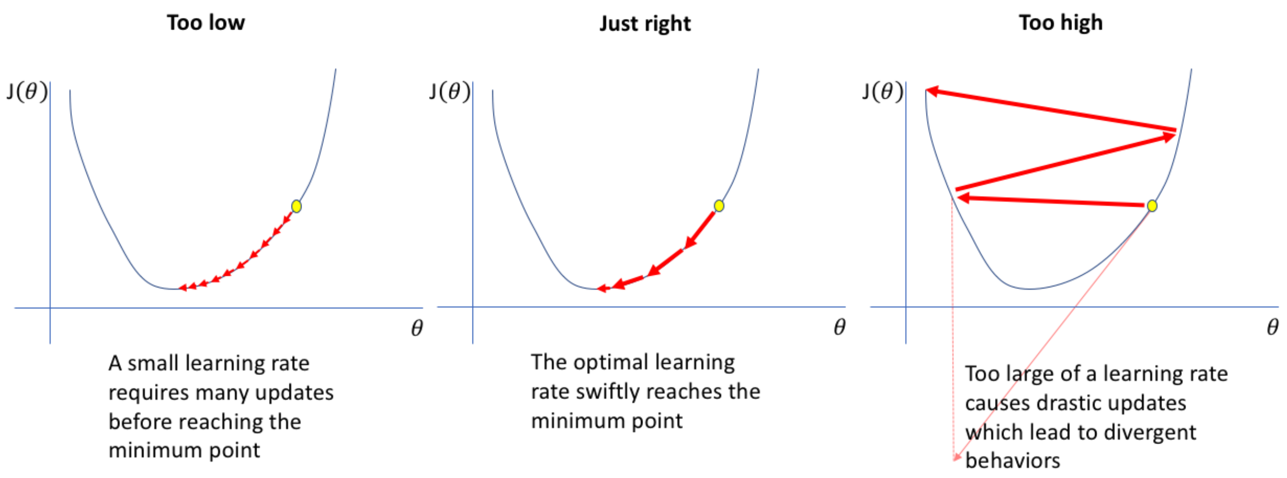
 end

until 模型 $f(\mathbf{x};\theta)$ 在验证集 $\mathcal{V}$ 上的错误率不再下降;

输出: θ

学习率是十分重要的超参数!





随机梯度下降法

- 随机梯度下降法（Stochastic Gradient Descent，SGD）也叫增量梯度下降，每个样本都进行更新

$$\theta_{t+1} = \theta_t - \alpha \frac{\partial \mathcal{L}(\theta_t; x^{(t)}, y^{(t)})}{\partial \theta},$$

- 小批量（Mini-Batch）随机梯度下降法

随机梯度下降法

算法 2.1: 随机梯度下降法

输入: 训练集 $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$, 验证集 $\mathcal{V}$, 学习率 α

随机初始化 θ;

repeat

  对训练集 $\mathcal{D}$ 中的样本随机重排序;

  for $n = 1 \cdots N$ **do**

    从训练集 $\mathcal{D}$ 中选取样本 $(\mathbf{x}^{(n)}, y^{(n)})$;

    // 更新参数

$$\theta \leftarrow \theta - \alpha \frac{\partial \mathcal{L}(\theta; x^{(n)}, y^{(n)})}{\partial \theta};$$

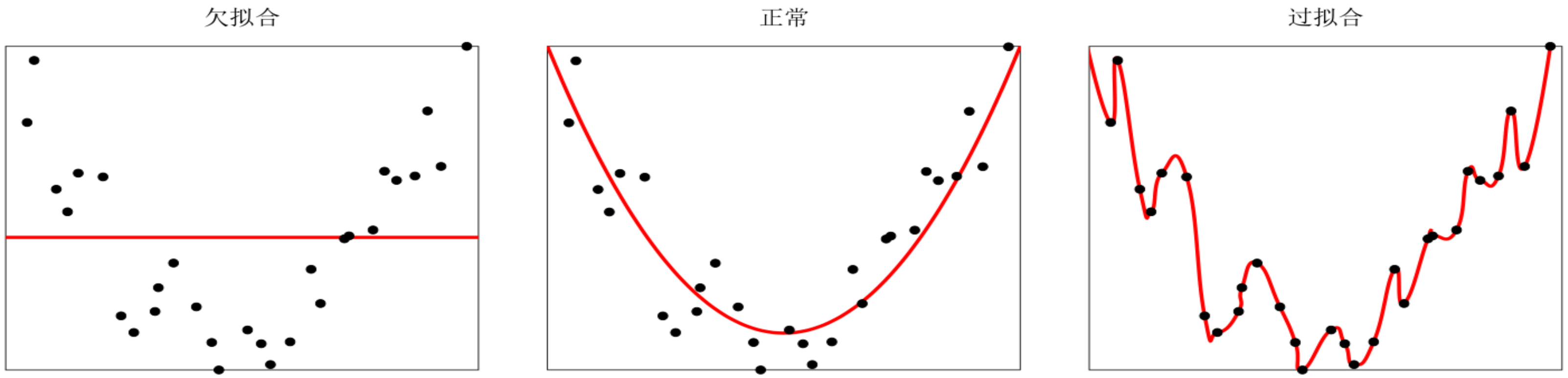
  end

until 模型 $f(\mathbf{x}; \theta)$ 在验证集 $\boxed{\mathcal{V}}$ 上的错误率不再下降;

输出: θ

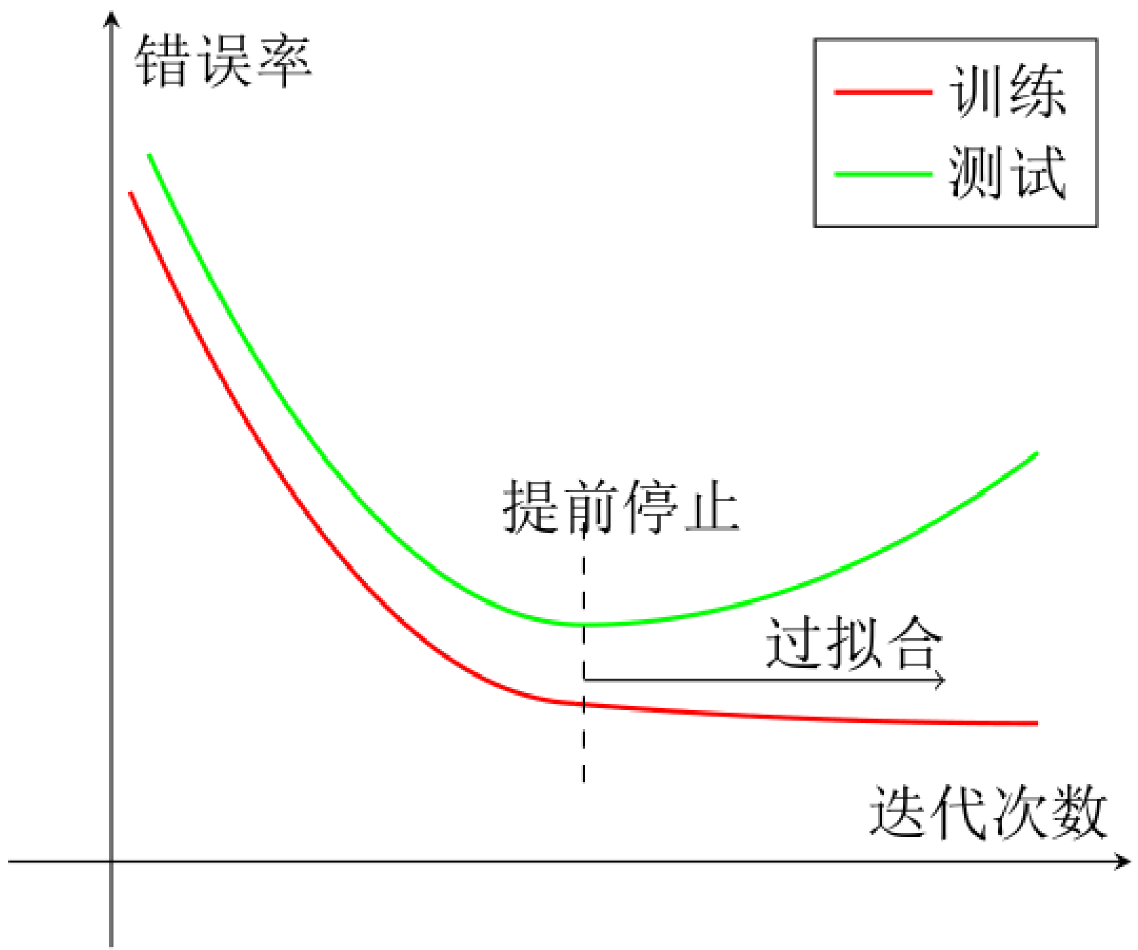
Why?

机器学习 = 优化?



过拟合：经验风险最小化原则很容易导致模型在训练集上错误率很低，但是在未知数据上错误率很高。

过拟合

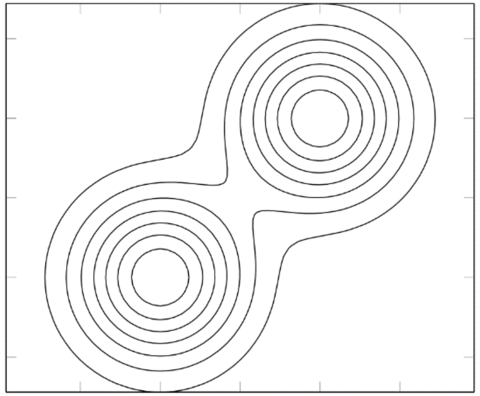


泛化错误

期望风险

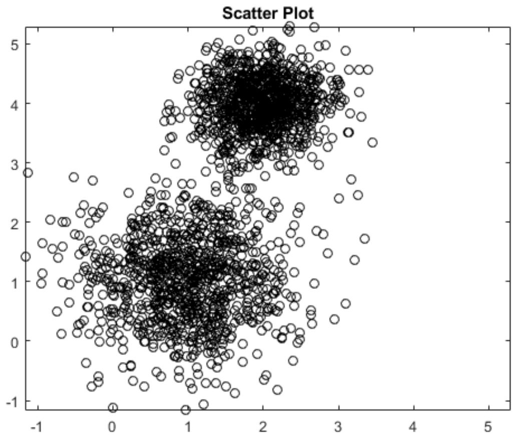
$$\mathcal{R}(f) = \mathbb{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}[\mathcal{L}(f(\mathbf{x}), y)],$$

真实分布 p_r



经验风险

$$\mathcal{R}_{\mathcal{D}}^{emp}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(y^{(n)}, f(x^{(n)}, \theta))$$



$\neq$

$$\mathcal{G}_{\mathcal{D}}(f) = \mathcal{R}(f) - \mathcal{R}_{\mathcal{D}}^{emp}(f)$$

泛化错误

如何减少泛化错误?

优化
经验风险最小

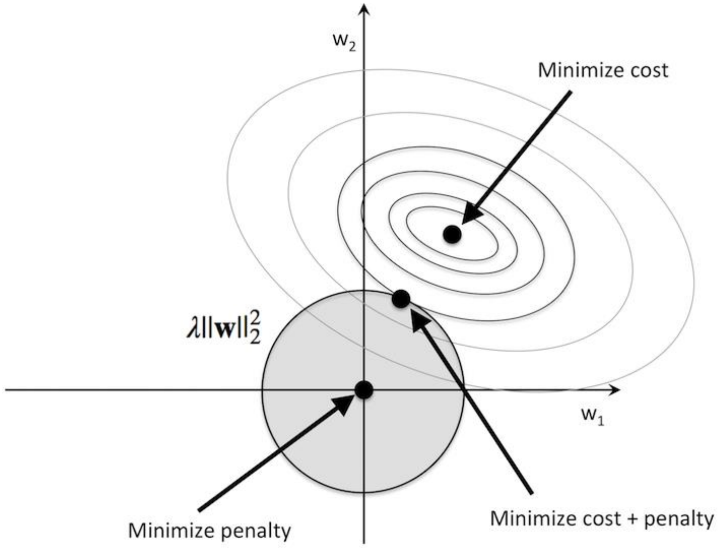
正则化
降低模型复杂度

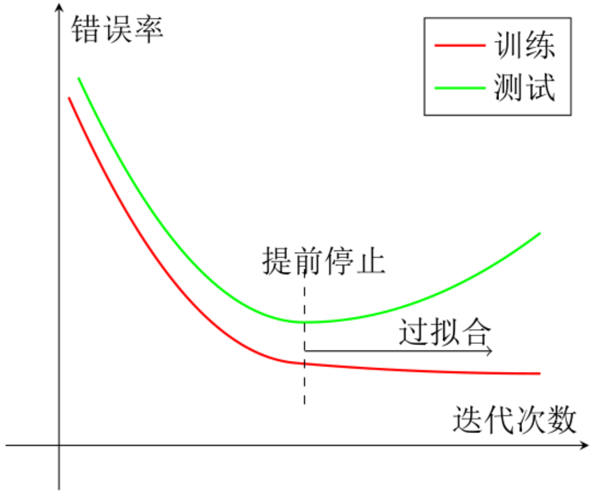
正则化（regularization）

所有损害优化的方法都是正则化。

结构风险最小化

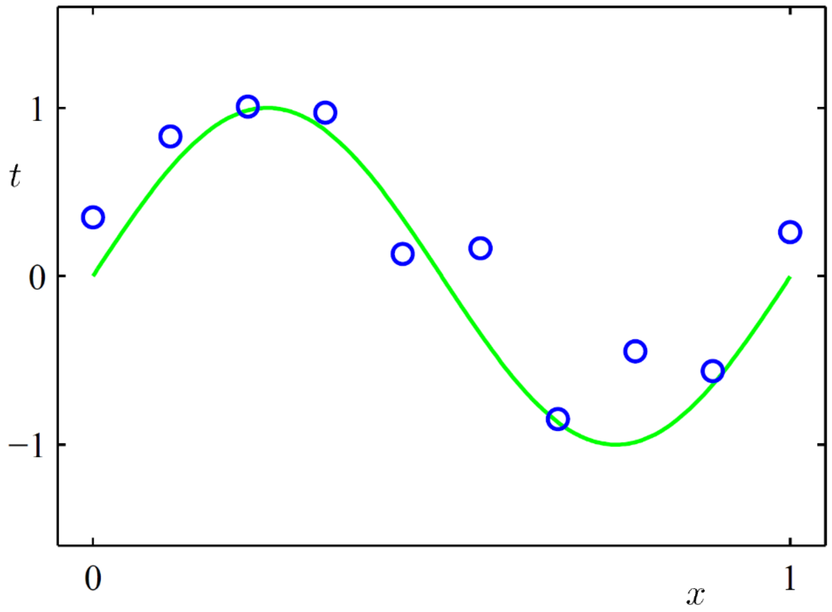
增加优化约束	干扰优化过程
L1/L2约束、数据增强	权重衰减、随机梯度下降、提前停止





多项式回归

一个例子：Polynomial Curve Fitting



模型 $$y(x, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M$$

损失函数 $$E(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \boldsymbol{w}) - t_n\}^2$$

经验风险最小化

模型

$$f(\mathbf{x};\mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}$$

学习准则

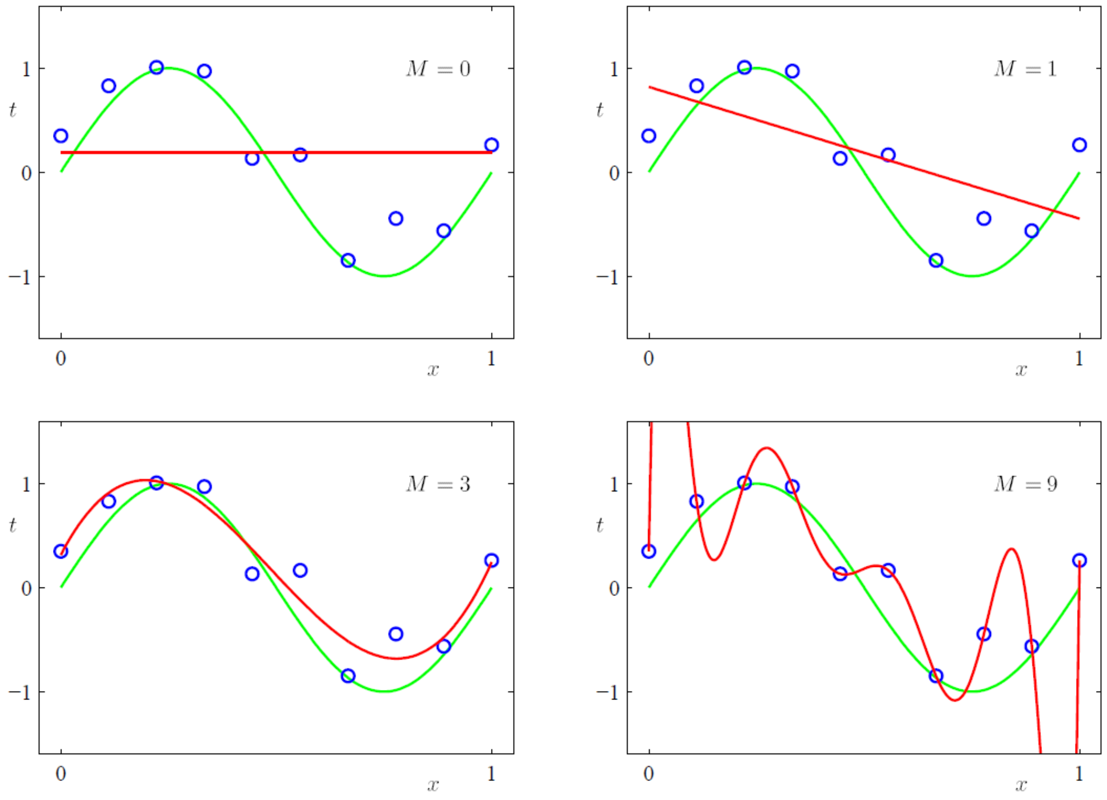
$$\begin{aligned}
\mathcal{R}(\mathbf{w}) &= \sum_{n=1}^{N} \mathcal{L}(y^{(n)}, f(\mathbf{x}^{(n)};\mathbf{w})) \\
&= \frac{1}{2}\sum_{n=1}^{N}\left(y^{(n)} - \mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)}\right)^2 \\
&= \frac{1}{2}\|\mathbf{y} - X^{\mathrm{T}}\mathbf{w}\|^2,
\end{aligned}$$

优化

$$\frac{\partial}{\partial \mathbf{w}}\mathcal{R}(\mathbf{w}) = 0$$

$$\begin{aligned}
\frac{\partial \mathcal{R}(\mathbf{w})}{\partial \mathbf{w}} &= \frac{1}{2}\frac{\partial \|\mathbf{y} - X^{\mathrm{T}}\mathbf{w}\|^2}{\partial \mathbf{w}} \\
&= -X(\mathbf{y} - X^{\mathrm{T}}\mathbf{w}),
\end{aligned}$$

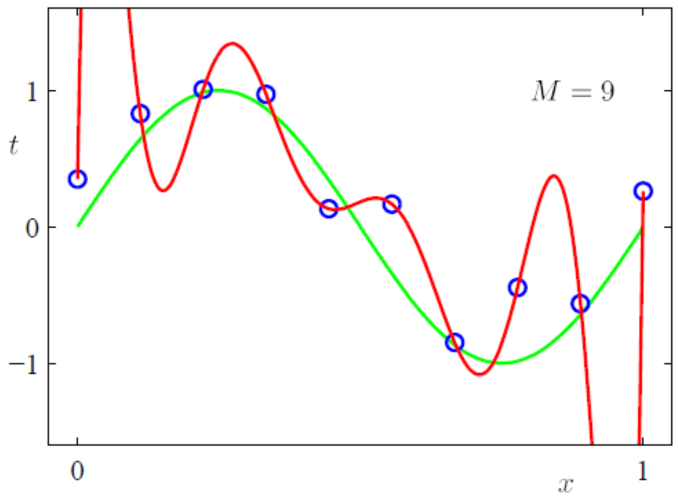
Which Degree of Polynomial?



A model selection problem

M = 9 → E(w) = 0: This is overfitting

Controlling Overfitting: Regularization



	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

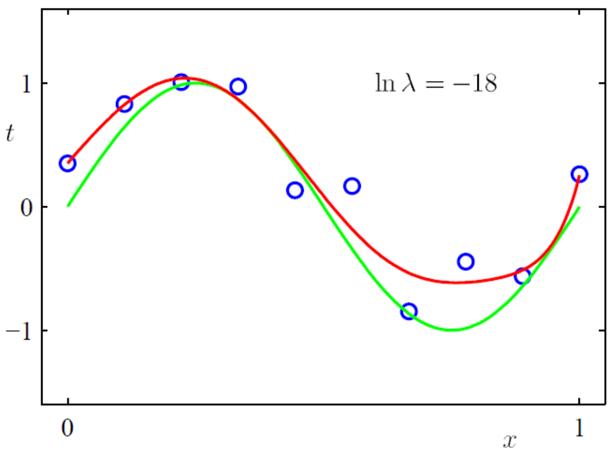
As order of polynomial M increases, so do coefficient magnitudes!

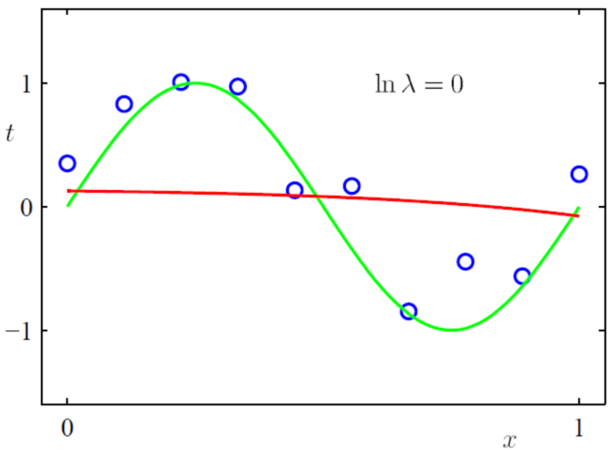
$$\tilde{E}(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\boldsymbol{w}) - t_n\}^2 + \frac{\lambda}{2}||\boldsymbol{w}||^2$$

对大的系数进行惩罚

Controlling Overfitting: Regularization

$$\tilde{E}(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \boldsymbol{w}) - t_n\}^2 + \frac{\lambda}{2}||\boldsymbol{w}||^2$$





	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

结构风险最小化

- 结构风险最小化准则

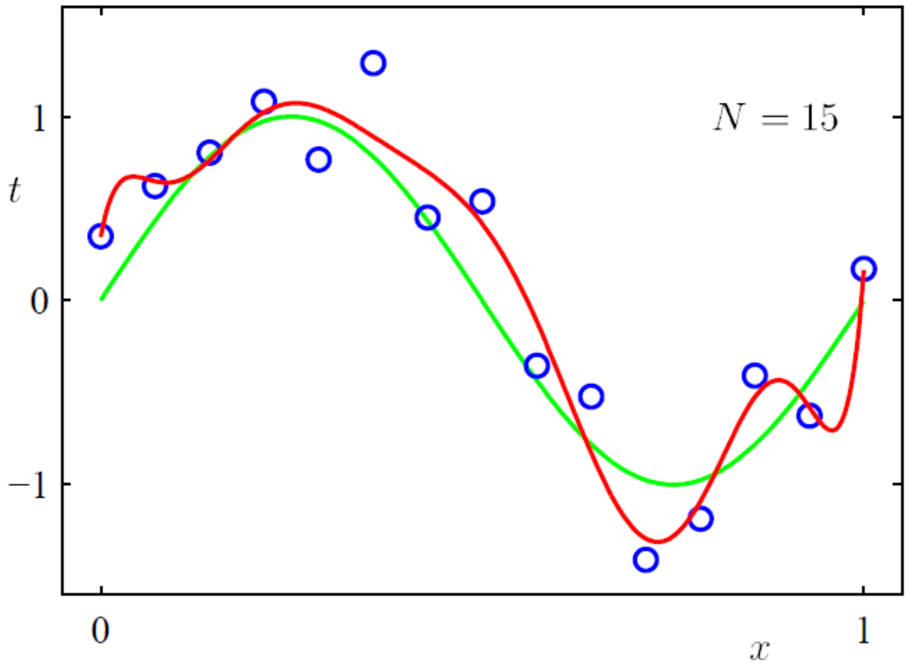
$$\mathcal{R}(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}^{\top}\boldsymbol{w}\|^2 + \frac{1}{2}\lambda\|\boldsymbol{w}\|^2,$$

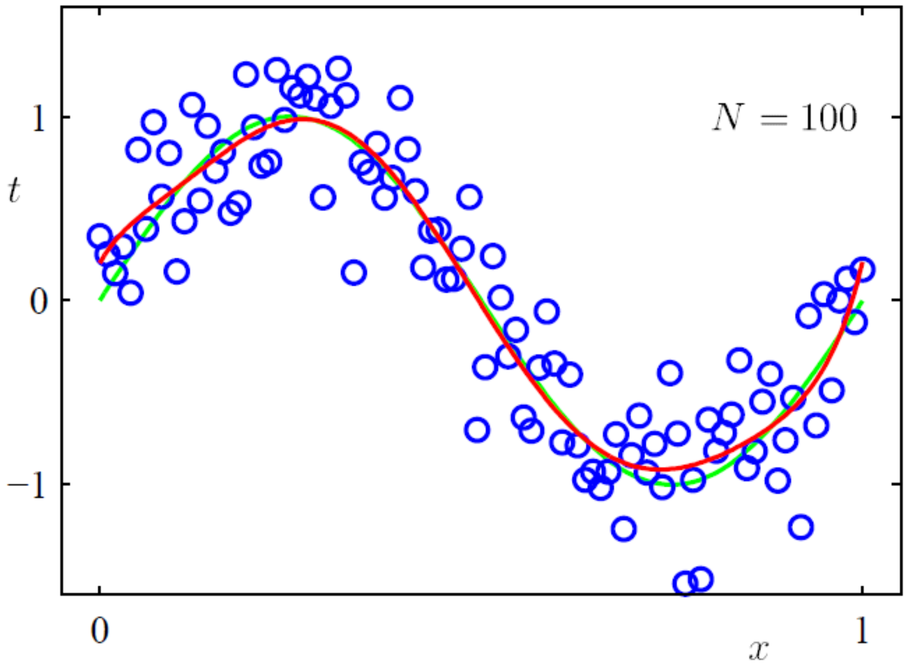
- 得到

$$\boldsymbol{w}^* = (\boldsymbol{X}\boldsymbol{X}^{\top} + \lambda I)^{-1}\boldsymbol{X}\boldsymbol{y},$$

- 岭回归（Ridge Regression）

Controlling Overfitting: Dataset size





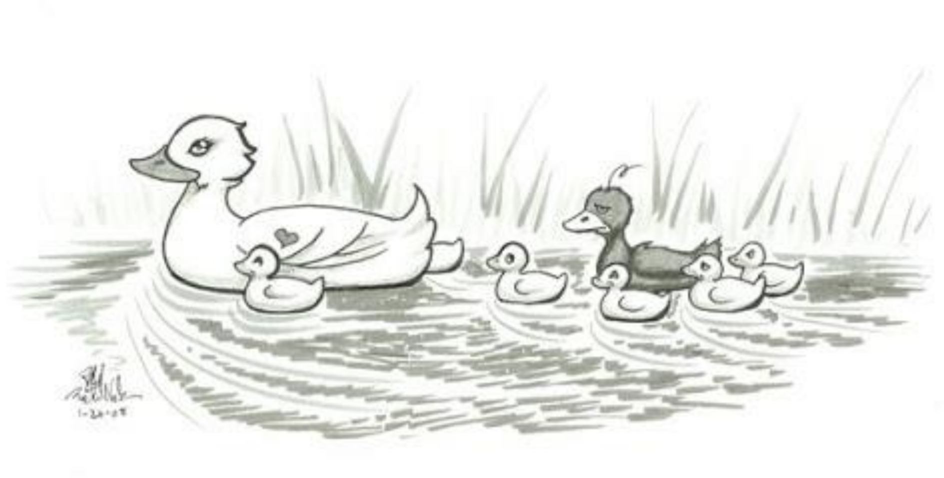
常用的定理

常用的定理

- 没有免费午餐定理（No Free Lunch Theorem，NFL）
  - 对于基于迭代的最优化算法，不存在某种算法对所有问题（有限的搜索空间内）都有效。如果一个算法对某些问题有效，那么它一定在另外一些问题上比纯随机搜索算法更差。

常用的定理

- 丑小鸭定理(Ugly Duckling Theorem)
  - 丑小鸭与白天鹅之间的区别和两只白天鹅之间的区别一样大.



常用的定理

- 奥卡姆剃刀原理(Occam's Razor)
  - 如无必要，勿增实体

课后作业

- 掌握知识点
  - 矩阵微分
  - 概率论
  - 信息论
  - 约束优化
- 编程练习
  - chp1_exercise