Machine Learning 2

B. Ensembles / B.3. Mixtures of Experts

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Computer Science
University of Hildesheim, Germany

Syllabus

			A. Advanced Supervised Learning
Fri.	12.4.	(1)	A.1 Generalized Linear Models
Fri.	26.4.	(2)	A.2 Gaussian Processes
Fri.	3.5.	(3)	A.2b Gaussian Processes (ctd.)
Fri.	10.5.	(4)	A.3 Advanced Support Vector Machines
			B. Ensembles
Fri.	17.5.	(5)	B.1 Stacking
Fri.	24.5.	(6)	B.2 Boosting
Fri.	31.5.	(7)	B.3 Mixtures of Experts
Fri.	7.6.	(8)	(ctd.)
Fri.	14.6.	—	— Pentecoste Break —
			C. Sparse Models
Fri.	21.6.	(9)	C.1 Homotopy and Least Angle Regression
Fri.	28.6.	(10)	C.2 Proximal Gradients
Fri.	29.6.	(11)	C.3 Laplace Priors & C.4 Automatic Relevance Determination
			D. Complex Predictors
Fri.	6.7.	(12)	D.1 Latent Dirichlet Allocation (LDA)
Fri.	12.7.	(13)	Q & A

Outline

1. The Idea behind Mixtures of Experts

2. Learning Mixtures of Experts

3. Interpreting Ensemble Models

Outline

1. The Idea behind Mixtures of Experts

2. Learning Mixtures of Experts

3. Interpreting Ensemble Models

Underlying Idea

So far, we build ensemble models where the combination weights do not depend on the predictors:

$$\hat{y}(x) := \sum_{c=1}^{C} \alpha_c \, \hat{y}_c(x)$$

i.e., all instances x are reconstructed from their predictions $\hat{y}_c(x)$ by the component models in the same way α.

Underlying Idea

So far, we build ensemble models where the combination weights do not depend on the predictors:

$$\hat{y}(x) := \sum_{c=1}^{C} \alpha_c \, \hat{y}_c(x)$$

i.e., all instances x are reconstructed from their predictions $\hat{y}_c(x)$ by the component models in the same way α.

New idea: allow each instance to be reconstructed in an instance-specific way.

$$\hat{y}(x) := \sum_{c=1}^{C} \alpha_c(x) \, \hat{y}_c(x)$$

Mixtures of Experts

$x_n \in \mathbb{R}^M, y_n \in \mathbb{R}, c_n \in \{1, \ldots, C\}, \theta := (\beta, \sigma^2, \gamma), \beta, \gamma \in \mathbb{R}^{C \times M}$:

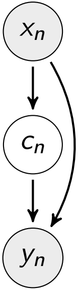
$$\begin{aligned} p(y_n \mid x_n, c_n; \theta) :=& \mathcal{N}(y \mid \beta_{c_n}^T x_n, \sigma_{c_n}^2) \\ p(c_n \mid x_n; \theta) :=& \text{Cat}(c \mid \mathcal{S}(\gamma x)) \end{aligned}$$

with **softmax function**

$$\mathcal{S}(x)_m := \frac{e^{x_m}}{\sum_{m'=1}^M e^{x_{m'}}}, \quad x \in \mathbb{R}^M$$

- C component models (**experts**) $\mathcal{N}(y \mid \beta_c^T x, \sigma_c^2)$
- each model c is expert in some region of predictor space, defined by its component weight (**gating function**) $\mathcal{S}(\gamma x)_c$
- a mixture model with latent nominal variable $z_n := c_n$.

Mixtures of Experts



$$x_n \in \mathbb{R}^M, y_n \in \mathbb{R}, c_n \in \{1, \ldots, C\}, \theta := (\beta, \sigma^2, \gamma), \beta, \gamma \in \mathbb{R}^{C \times M}:$$

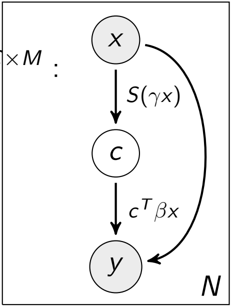
$$p(y_n \mid x_n, c_n; \theta) := \mathcal{N}(y \mid \beta_{c_n}^T x_n, \sigma_{c_n}^2)$$
$$p(c_n \mid x_n; \theta) := \text{Cat}(c \mid \mathcal{S}(\gamma x))$$

with **softmax function**

$$\mathcal{S}(x)_m := \frac{e^{x_m}}{\sum_{m'=1}^M e^{x_{m'}}}, \quad x \in \mathbb{R}^M$$

- C component models (**experts**) $\mathcal{N}(y \mid \beta_c^T x, \sigma_c^2)$
- each model c is expert in some region of predictor space, defined by its component weight (**gating function**) $\mathcal{S}(\gamma x)_c$
- a mixture model with latent nominal variable $z_n := c_n$.

Mixtures of Experts



$x_n \in \mathbb{R}^M, y_n \in \mathbb{R}, c_n \in \{1, \ldots, C\}, \theta := (\beta, \sigma^2, \gamma), \beta, \gamma \in \mathbb{R}^{C \times M}$:

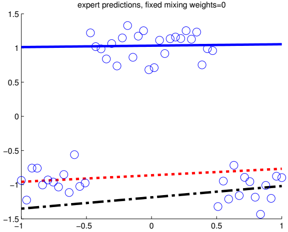
$$p(y_n \mid x_n, c_n; \theta) := \mathcal{N}(y \mid \beta_{c_n}^T x_n, \sigma_{c_n}^2)$$
$$p(c_n \mid x_n; \theta) := \text{Cat}(c \mid \mathcal{S}(\gamma x))$$

with **softmax function**

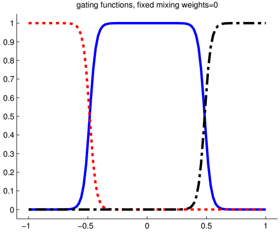
$$\mathcal{S}(x)_m := \frac{e^{x_m}}{\sum_{m'=1}^M e^{x_{m'}}}, \quad x \in \mathbb{R}^M$$

- C component models (**experts**) $\mathcal{N}(y \mid \beta_c^T x, \sigma_c^2)$
- each model c is expert in some region of predictor space, defined by its component weight (**gating function**) $\mathcal{S}(\gamma x)_c$
- a mixture model with latent nominal variable $z_n := c_n$.

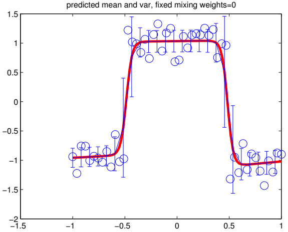
Mixtures of Experts/ Example



component models



component weight



mixture of experts

[Mur12, fig. 11.6]

Mixtures of Experts

Generic Mixtures of Experts model:

- variables: $x_n \in \mathcal{X}, y_n \in \mathcal{Y}$
- latent variables: $c_n \in \{1, \ldots, C\}$
- **component models**: $p(y_n \mid x_n, c_n; \theta^y)$
  - a separate model for each c: $p(y_n \mid x_n, c; \theta^y) = p(y_n \mid x_n; \theta^y_c)$, with θ^y_c and $\theta^y_{c'}$ being disjoint for $c \neq c'$.
- **combination model**: $p(c_n \mid x_n; \theta^c)$

Example Mixture of Experts model:

- variables: $\mathcal{X} := \mathbb{R}^M, \mathcal{Y} := \mathbb{R}$
- component models: linear regression models $\mathcal{N}(y \mid \beta_c^T x, \sigma_c^2)$
- combination model: logistic regression model $\text{Cat}(c \mid \mathcal{S}(\gamma x))$

For prediction:

$$p(y \mid x) = \sum_{c=1}^{C} p(y \mid x, c)\, p(c \mid x)$$

Mixtures of Experts

Generic Mixtures of Experts model:

- variables: $x_n \in \mathcal{X}, y_n \in \mathcal{Y}$
- latent variables: $c_n \in \{1, \ldots, C\}$
- **component models**: $p(y_n \mid x_n, c_n; \theta^y)$
  - a separate model for each c: $p(y_n \mid x_n, c; \theta^y) = p(y_n \mid x_n; \theta^y_c)$, with θ^y_c and $\theta^y_{c'}$ being disjoint for $c \neq c'$.
- **combination model**: $p(c_n \mid x_n; \theta^c)$

Example Mixture of Experts model:

- variables: $\mathcal{X} := \mathbb{R}^M, \mathcal{Y} := \mathbb{R}$
- component models: linear regression models $\mathcal{N}(y \mid \beta_c^T x, \sigma_c^2)$
- combination model: logistic regression model $\text{Cat}(c \mid \mathcal{S}(\gamma x))$

For prediction:

$$p(y \mid x) = \sum_{c=1}^{C} \underbrace{p(y \mid x, c)}_{=\hat{y}_c(x)} \underbrace{p(c \mid x)}_{=\alpha_c(x)}$$

Outline

Learning Mixtures of Experts

complete data likelihood:

$$\ell(\theta^y, \theta^c, c; \mathcal{D}^{\text{train}}) := \prod_{n=1}^{N} p(y_n | x_n, c_n; \theta^y) p(c_n | x_n; \theta^c), \quad c_n \in \{1, \ldots, C\}$$

Cannot be computed, as c_n is unknown.

Learning Mixtures of Experts

complete data likelihood:

$$\ell(\theta^y, \theta^c, c; \mathcal{D}^{\text{train}}) := \prod_{n=1}^{N} p(y_n | x_n, c_n; \theta^y) p(c_n | x_n; \theta^c), \quad c_n \in \{1, \ldots, C\}$$

Cannot be computed, as c_n is unknown.

weighted complete data likelihood:

$$\ell(\theta^y, \theta^c, w; \mathcal{D}^{\text{train}}) := \prod_{n=1}^{N} \prod_{c=1}^{C} \left(p(y_n | x_n, c; \theta^y) p(c | x_n; \theta^c)\right)^{w_{n,c}}, \quad w_n \in \Delta_C$$

$$-\log \ell(\theta^y, \theta^c, w; \mathcal{D}^{\text{train}}) = -\sum_{n=1}^{N} \sum_{c=1}^{C} w_{n,c} \left(\log p(y_n | x_n, c; \theta^y) + \log p(c | x_n; \theta^c)\right)$$

Note: $\Delta_C := \{w \in [0,1]^C \mid \sum_{c=1}^{C} w_c = 1\}$.

Learning Mixtures of Experts

complete data likelihood:

$$\ell(\theta^y, \theta^c, c; \mathcal{D}^{\text{train}}) := \prod_{n=1}^{N} p(y_n | x_n, c_n; \theta^y) p(c_n | x_n; \theta^c), \quad c_n \in \{1, \ldots, C\}$$

Cannot be computed, as c_n is unknown.

weighted complete data likelihood:

$$\ell(\theta^y, \theta^c, w; \mathcal{D}^{\text{train}}) := \prod_{n=1}^{N} \prod_{c=1}^{C} (p(y_n | x_n, c; \theta^y) p(c | x_n; \theta^c))^{w_{n,c}}, \quad w_n \in \Delta_C$$

$$-\log \ell(\theta^y, \theta^c, w; \mathcal{D}^{\text{train}}) = -\sum_{n=1}^{N} \sum_{c=1}^{C} w_{n,c} (\log p(y_n | x_n, c; \theta^y) + \log p(c | x_n; \theta^c))$$

Cannot be computed either, as w_n is unknown;
but w_n can be treated as parameter — but with unwanted consequences.

Note: $\Delta_C := \{w \in [0,1]^C \mid \sum_{c=1}^{C} w_c = 1\}$.

Learning Mixtures of Experts

If we treat w_n as free parameters, two issues emerge:

1. their **relation to θ^y and θ^c** via
$$w_{n,c} \overset{!}{=} p(c_n = c \mid x_n, y_n; \theta^y, \theta^c) \underset{\text{Bayes}}{=} \frac{p(y_n \mid x_n, c; \theta^y) p(c \mid x_n; \theta^c)}{\sum_{c'=1}^{C} p(y_n \mid x_n, c'; \theta^y) p(c' \mid x_n; \theta^c)}$$
is not modeled.
2. a block coordinate descent approach for $w_{n,c}$ would it set trivially to **crisp estimates**:
$$\arg\min_{w_{1:N,1:C}} - \sum_{n=1}^{N} \sum_{c=1}^{C} w_{n,c} \overbrace{(\log p(y_n \mid x_n, c; \theta^y) + \log p(c \mid x_n; \theta^c))}^{=: a_{n,c}},$$
decomposes over n:
$$\forall n: \quad \arg\min_{w_{n,1:C}} - \sum_{c=1}^{C} w_{n,c} a_{n,c}, \quad w_n \in \Delta_C$$
$$\rightsquigarrow \quad w_{n,c} := \mathbb{I}(c = \arg\max a_{n,c'})$$

Learning Mixtures of Experts / Bilevel Optimization

Formulate the problem as bilevel optimization problem:

$$(\theta^y, \theta^c) := \arg\max_{\theta^y, \theta^c} \ell(\theta^y, \theta^c; \mathcal{D}^{\text{train}})$$

$$:= \prod_{n=1}^{N} \prod_{c=1}^{C} (p(y_n \mid x_n, c; \theta^y) p(c \mid x_n; \theta^c))^{w_{n,c}(\theta^y, \theta^c)}$$

with

$$w_{n,1:C}(\theta^y, \theta^c) := \arg\min_{w_{n,1:C}} \sum_{c=1}^{C} \left(\frac{p(y_n \mid x_n, c; \theta^y) p(c \mid x_n; \theta^c)}{\sum_{c'=1}^{C} p(y_n \mid x_n, c'; \theta^y) p(c' \mid x_n; \theta^c)} - w_{n,c} \right)^2$$

$$\forall n = 1 : N$$

Generic **bilevel optimization problem**:

$$x := \arg\min_x f(x, y(x))$$

$$\text{with } y(x) := \arg\min_y g(x, y)$$

Bilevel Optimization

Generic **bilevel optimization problem**:

$$x := \arg\min_x f(x, y(x)) \qquad \text{outer problem}$$

$$\text{with } y(x) := \arg\min_y g(x, y) \qquad \text{inner problem}$$

argmin-bilevel-alternate$(f, g, x^{(0)}, \epsilon)$:
- $t := 0$
- do
  - $t := t + 1$
  - $y^{(t)} := \arg\min_y g(x^{(t-1)}, y)$
  - $x^{(t)} := \arg\min_x f(x, y^{(t)})$
- while $||x^{(t)} - x^{(t-1)}|| < \epsilon$
- return $x^{(t)}$

- convergence in general problematic

Learning Mixtures of Experts

$$\arg\min_{\theta^y,\theta^c} -\sum_{n=1}^{N}\sum_{c=1}^{C} w_{n,c}(\theta^y,\theta^c)\left(\log p(y_n|x_n,c;\theta^y)+\log p(c|x_n;\theta^c)\right)$$

$$w_{n,1:C}(\theta^y,\theta^c) := \arg\min_{w_{n,1:C}} \sum_{c=1}^{C}\left(\frac{p(y_n \mid x_n, c;\theta^y)p(c \mid x_n;\theta^c)}{\sum_{c'=1}^{C} p(y_n \mid x_n, c';\theta^y)p(c' \mid x_n;\theta^c)} - w_{n,c}\right)^2$$

Strategy:

- alternate inner/outer for bilevel (EM)
- Block coordinate descent for outer problem:

Learning Mixtures of Experts

$$\arg\min_{\theta^y,\theta^c} -\sum_{n=1}^{N}\sum_{c=1}^{C} w_{n,c}(\theta^y,\theta^c)\left(\log p(y_n|x_n,c;\theta^y)+\log p(c|x_n;\theta^c)\right)$$

$$w_{n,1:C}(\theta^y,\theta^c) := \arg\min_{w_{n,1:C}} \sum_{c=1}^{C}\left(\frac{p(y_n \mid x_n,c;\theta^y)p(c \mid x_n;\theta^c)}{\sum_{c'=1}^{C} p(y_n \mid x_n,c';\theta^y)p(c' \mid x_n;\theta^c)} - w_{n,c}\right)^2$$

Strategy:

- alternate inner/outer for bilevel (EM)
- Block coordinate descent for outer problem:

1. minimize inner problem w.r.t. $w_{n,c}$:
   - decomposes into $N \cdot C$ problems
   - analytic solution:

$$w_{n,c} = \frac{p(y_n \mid x_n,c;\theta^y)p(c \mid x_n;\theta^c)}{\sum_{c'=1}^{C} p(y_n \mid x_n,c';\theta^y)p(c' \mid x_n;\theta^c)}$$

Learning Mixtures of Experts

$$\arg\min_{\theta^y,\theta^c} - \sum_{n=1}^{N}\sum_{c=1}^{C} w_{n,c}(\theta^y,\theta^c)\left(\log p(y_n|x_n,c;\theta^y) + \log p(c|x_n;\theta^c)\right)$$

$$w_{n,1:C}(\theta^y,\theta^c) := \arg\min_{w_{n,1:C}} \sum_{c=1}^{C}\left(\frac{p(y_n \mid x_n,c;\theta^y)p(c \mid x_n;\theta^c)}{\sum_{c'=1}^{C} p(y_n \mid x_n,c';\theta^y)p(c' \mid x_n;\theta^c)} - w_{n,c}\right)^2$$

Strategy:

- alternate inner/outer for bilevel (EM)
- Block coordinate descent for outer problem:

2. minimize outer problem w.r.t. θ^y:
   - decomposes into C problems $\arg\min_{\theta_c^y} - \sum_{n=1}^{N} w_{n,c}\log p(y_n|x_n;\theta_c^y)$
   - learn C component models for $\mathcal{D}^{\text{train}}$ with case weights $w_{n,c}$.

Learning Mixtures of Experts

$$\arg\min_{\theta^y,\theta^c} -\sum_{n=1}^{N}\sum_{c=1}^{C} w_{n,c}(\theta^y,\theta^c)\left(\log p(y_n|x_n,c;\theta^y)+\log p(c|x_n;\theta^c)\right)$$

$$w_{n,1:C}(\theta^y,\theta^c) := \arg\min_{w_{n,1:C}} \sum_{c=1}^{C}\left(\frac{p(y_n \mid x_n,c;\theta^y)p(c \mid x_n;\theta^c)}{\sum_{c'=1}^{C} p(y_n \mid x_n,c';\theta^y)p(c' \mid x_n;\theta^c)} - w_{n,c}\right)^2$$

Strategy:

- alternate inner/outer for bilevel (EM)
- Block coordinate descent for outer problem:

3. minimize outer problem w.r.t. θ^c:
   - solve
     $$\arg\min_{\theta^c} -\sum_{n=1}^{N}\sum_{c=1}^{C} w_{n,c}\log p(c|x_n;\theta^c)$$
   - learn a combination model for target c on
     $$\mathcal{D}^{\text{train,wcompl}} := \{(x_n,c,w_{n,c}) \mid n=1,\ldots,N, c=1,\ldots,C\}$$

Remarks

- Mixtures of experts can use **any model as component model**.
- Mixtures of experts can use **any classification model as combination model**.
  - both models need to be able to **deal with case weights**
  - both models need to be able to **output probabilities**

Remarks

- Mixtures of experts can use **any model as component model**.
- Mixtures of experts can use **any classification model as combination model**.
  - both models need to be able to **deal with case weights**
  - both models need to be able to **output probabilities**

- if data is **sparse**, sparsity can be naturally used in both, component and combination models.

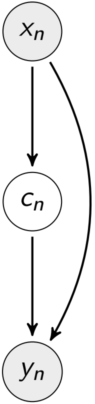
Remarks

- Mixtures of experts can use **any model as component model**.
- Mixtures of experts can use **any classification model as combination model**.
  - both models need to be able to **deal with case weights**
  - both models need to be able to **output probabilities**

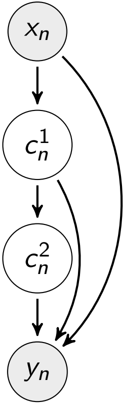
- if data is **sparse**, sparsity can be naturally used in both, component and combination models.

- Updating the three types of parameters can be **interleaved**.
  - this way, $w_{n,c}$ never has to be materialized (but for a mini batch, possibly a single n)

Outlook: Hierarchical Mixture of Experts



mixture of experts



hierarchical mixture of experts

Outline

Variable Importance

Some models allow to assess the importance of single variables
(or more generally subsets of variables; **variable importance**), e.g.,

- linear models: the z-score
- decision trees: the number of times a variable occurs in its splits

Variable Importance

Some models allow to assess the importance of single variables (or more generally subsets of variables; **variable importance**), e.g.,

- linear models: the z-score
- decision trees: the number of times a variable occurs in its splits

Variable importance of ensembles of such models can be measured as **average variable importance in the component models**:

$$\text{importance}(X_m, \hat{y}) := \frac{1}{C} \sum_{c=1}^{C} \text{importance}(X_m, \hat{y}_c), \quad m \in \{1, \ldots, M\}$$

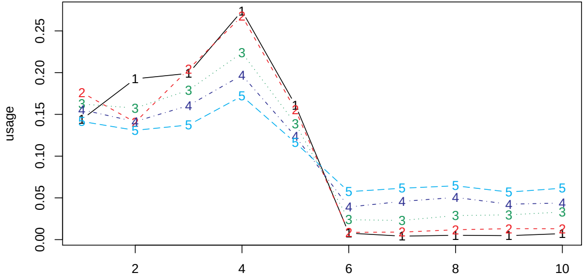
Variable Importance / Example

Synthetic data:

$$x \sim \text{uniform}([0,1]^{10})$$

$$y \sim \mathcal{N}(y \mid 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5, 1)$$

Model: Bayesian adaptive regression tree
(variant of a random forest; see [Mur12, p. 551]).



[Mur12, fig. 16.21]

Color denotes the number C of component models.

Variable Dependence: Partial Dependence Plot

For any model $\hat{y}$ (and thus any ensemble), the dependency of the model on a variable X_m can be visualized by a **partial dependence plot**:

plot $z \in \text{range}(X_m)$ vs.

$$\hat{y}_{\text{partial}}(z; X_m, \mathcal{D}^{\text{train}}) := \frac{1}{N} \sum_{n=1}^{N} \hat{y}((x_{n,1}, \ldots, x_{n,m-1}, z, x_{n,m+1}, \ldots, x_{n,M})),$$

or for a subset of variables

$$\hat{y}_{\text{partial}}(z; X_V, \mathcal{D}^{\text{train}}) := \frac{1}{N} \sum_{n=1}^{N} \hat{y}(\rho(x, V, z)), \quad V \subseteq \{1, \ldots, M\}$$

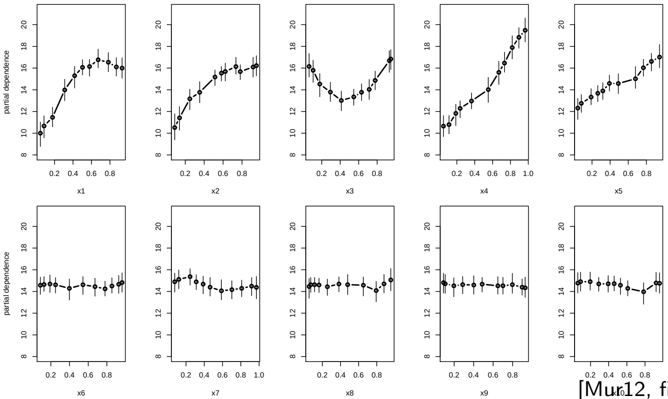
$$\text{with } \rho(x, V, z)_m := \begin{cases} z_m, & \text{if } m \in V \\ x_m, & \text{else} \end{cases}, \quad m \in \{1, \ldots, M\}$$

Variable Dependence / Example

Synthetic data:

$$x \sim \text{uniform}([0,1]^{10})$$

$$y \sim \mathcal{N}(y \mid 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10 x_4 + 5 x_5, 1)$$



[Mur12, fig. 16.20]

Summary

- **Mixtures of Experts** additionally allow the combination weights to depend on x (**gating function**)
  - jointy model
    - a latent component each instance belongs to and
    - a model for y for each component
  - can be learned via block coordinate descent / EM.
    - requiring just learning algorithms for the component models
    - as well as for the combination model.

- Ensemble models can be diagnosed by **partial dependence plots** (as any model).

Further Readings

- Mixtures of Experts: [Bis06, chapter 14.5]. [Mur12, chapter 11.2.4, 11.4.3], [HTFF05, chapter 9.5].
- Bilevel optimization:
  - an interesting application of bilevel optimization in ML for hyperparameter optimization: [FFS+18].

Acknowledgements: Thanks a lot to my PhD student Randolf Scholz for spotting a bad mistake on an earlier version of these slides!

References

Christopher M. Bishop.
Pattern recognition and machine learning, volume 1.
springer New York, 2006.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil.
Bilevel Programming for Hyperparameter Optimization and Meta-Learning.
In *International Conference on Machine Learning*, pages 1568–1577, July 2018.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin.
The elements of statistical learning: data mining, inference and prediction, volume 27.
Springer, 2005.

Kevin P. Murphy.
Machine learning: a probabilistic perspective.
The MIT Press, 2012.