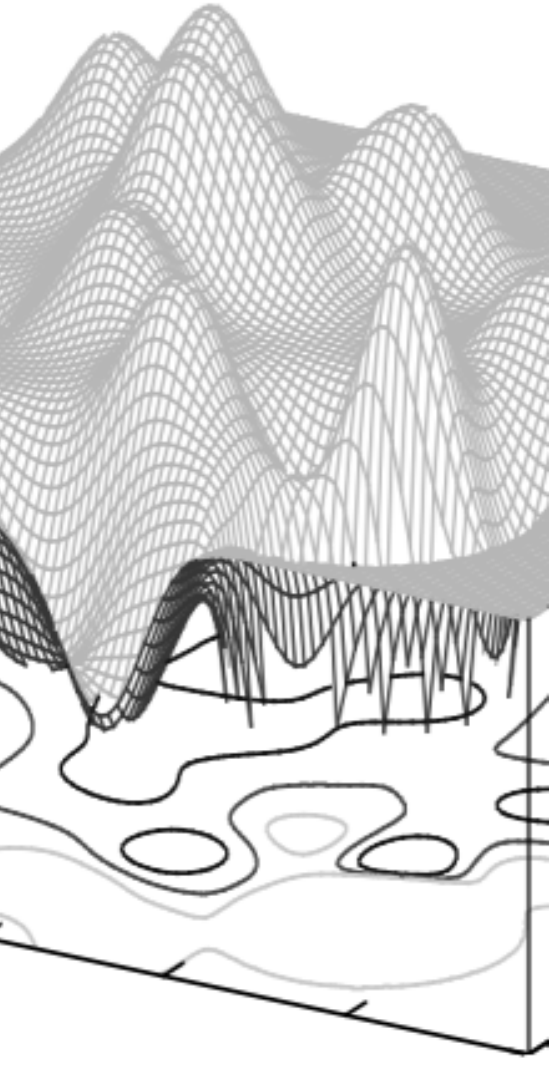

Machine Learning

Recap

Philipp Kratzer & Janik Hager

Marc Toussaint
University of Stuttgart
Summer 2019

What is Machine Learning?

- Pedro Domingos: *A Few Useful Things to Know about Machine Learning*

$$\text{learning} = \text{representation} + \text{evaluation} + \text{optimization}$$

- “Representation”: Choice of model, choice of hypothesis space
- “Evaluation”: Choice of objective function, optimality principle
- “Optimization”: The algorithm to compute/approximate the best model

Regression: Ridge Regression

- **Representation:** choice of features

$$f(x) = \phi(x)^\top \beta$$

- **Objective:** squared error + Ridge/Lasso regularization

$$L^{\text{ridge}}(\beta) = \sum_{i=1}^{n} (y_i - \phi(x_i)^\top \beta)^2 + \lambda \|\beta\|_I^2$$

- **Solver:** analytical (or quadratic program for Lasso)

$$\hat{\beta}^{\text{ridge}} = (X^\top X + \lambda I)^{-1} X^\top y$$

Classification: Logistic Regression

- **Representation:** choice of features

$$f(x) = \phi(x, y)^\top \beta$$

- **Objective:** neg-log-likelihood

$$L^{\text{logistic}}(\beta) = -\sum_{i=1}^{n} \log p(y_i \mid x_i) + \lambda \|\beta\|^2$$

$$p(y|x) \propto e^{f(x,y)}$$

- **Solver:** numerical (Newton algorithm)

$$\beta \leftarrow \beta - (X^\top W X + 2\lambda I)^{\text{-}1} \, (X^\top (p - y) + 2\lambda I \beta)$$

Neural Networks

- **Representation:** multi-layer, sequential mapping

$$f(x) = W_2\sigma(W_1\sigma(W_0x + b_0) + b_1) + b_2$$

- **Objective:** e.g. a squared loss for regression

$$L(f) = \sum_{i=1}^{n}(y_i - f(x_i))^2$$

- **Solver:** Propagating the error backwards, while compute the gradients $\frac{dL(f)}{dW_l}$ for each layer. Weight update can be done using e.g. stochastic gradient descent ($\beta = (W_{1:L}, b_{1:L})$)

$$\beta \leftarrow \beta - \eta\nabla_\beta L(\beta, \hat{D})$$

Neural Networks

- **activation functions:** ReLU, leaky ReLU, sigmoid, ...
- **regularization:** dropout, data augmentation, early stopping, ...
- **special NNs:** Convolutional NNs (images), LSTM (time series), ...

Kernelization

- **Representation:** Kernel Ridge Regression

$$f^{\text{rigde}}(x) = \kappa(x)^\top (K + \lambda I)^{-1} y$$
$$\text{with } K_{ij} = k(x_i, x_j)$$
$$\kappa_i(x) = k(x, x_i)$$

- **Kernel:** Every choice of features implies a kernel and the other way round.

$$k(x_i, x_j) = \phi(x_i)^\top \phi(x_j)$$

Unsupervised Learning: PCA

$V_p^\top$ is the matrix that projects to the largest variance directions of $\tilde{X}^\top \tilde{X}$

- **Representation:**

$$x \approx V_p z + \mu$$

- **Objective:**

$$\sum_{i=1}^{n} \|x_i - (V_p z_i + \mu)\|^2$$

- **Solver:** Eigenvector decomposition of $\tilde{X}^\top \tilde{X}$

Unsupervised Learning: Clustering

k-means:

- **Representation:** K centers μ_k and a data assignment $c : i \mapsto k$
- **Objective:**

$$\min_{c,\mu} \sum_i (x_i - \mu_{c(i)})^2$$

- **Solver:**
  - Pick K data points randomly to initialize the centers μ_k
  - Iterate adapting the assignments $c(i)$ and the centers μ_k

Gaussian Mixture Models:

Approximate the ”true” distribution, from which the data $\{x_i\}_{i=1}^N$ is generated, using a mixture of multivariate Gaussians (solved via EM-Algorithm).

Local Learning & Combining Models

- **Local Learning:** Build local model using kNN of query x^*
- **Model Averaging:** Fully different types of models (using different (e.g. limited) feature sets; neural nets; decision trees; hyperparameters)
- **Bootstrap:** Models of same type, trained on randomized versions of D
- **Boosting:** Models of same type, trained on cleverly designed modifications/reweightings of D
- **How to choose weights for combining models:**
  naive averaging, Bayesian Model Averaging, Function view, ...

Bayesian Models

Placing distributions on parameters, model classes, ...

- **Representation:** e.g. Kernel Bayesian Logistic Regression

$$P(X), \quad P(\beta), \quad P(Y|X,\beta)$$

- **Objective & Solver:** compute inference

$$P(\beta \,|\, x_{1:n}, y_{1:n}) = \frac{\prod_{i=1}^n P(y_i \,|\, \beta, x_i)\; P(\beta)}{Z}$$

- **Insights:**
  - The *neg-log posterior* $P(\beta \,|\, D)$ is proportional to the cost function $L^{\text{ridge}}(\beta)$.
  - The mean $\hat{\beta}$ is exactly the classical $\operatorname{argmin}_\beta L^{\text{ridge}}(\beta)$.
  - The Bayesian inference approach not only gives a mean/optimal $\hat{\beta}$, but also a variance Σ of that estimate.

Summary

- Machine Learning is a large field with many real-world applications
- Includes many components from computer science and statistics
- Further points covered in the lecture: tree-based models, conditional random fields, ...

Summary

- Machine Learning is a large field with many real-world applications
- Includes many components from computer science and statistics
- Further points covered in the lecture: tree-based models, conditional random fields, ...

  All models are wrong, but some are useful.
  George Box, 1919 - 2013