CS489/698
Lecture 23: March 29, 2017

Stream learning, course wrap up

[M] Sec. 8.5

Stream Learning

- Classic machine learning
  - Batch learning: fixed dataset
  - Train once on dataset
- **Stream learning**
  - **Online learning: new data is continuously arriving**
  - Continuously train as new data arrives
- Applications of stream learning
  - Recommender systems (e.g., movie and app recommendations)
  - Time series prediction (e.g., weather, stock market)
  - Big data (e.g. process dataset sequentially)

Streaming challenges

- Since the data is streaming, we can’t store it all
- The learning algorithm must keep up with the stream
- Data patterns may change over time

- Stream learning: learner must be able to take a hypothesis as input and update it each time a new data point (or mini-batch of data points) arrive.
  - Cannot revisit older data (since we can’t store it all)
  - Time to process new data point (or mini-batch of data points) must be constant and less than the arrival time for the next data point (or mini-batch of data points).

Bayesian Learning

- Examples: Bayesian linear regression, Gaussian processes
- Bayesian learning lends itself naturally to stream/online learning.

- Bayes theorem:

Optimization Based Learning

- Many ML algorithms are based on optimization: least square regression, logistic regression, maximum likelihood, support vector machines, neural networks

- How do we devise an incremental optimization algorithm that looks at each data point just once?

Optimization-based Learning

- Optimization based ML algorithms are typically formulated as follows:

$$\theta^* = argmin_\theta \; Loss(data; \theta)$$
$$= argmin_\theta \sum_n Loss(x_n, y_n; \theta)$$

Where $Loss(x, y; \theta)$ might be

- negative log likelihood: $-\log \Pr(y|x; \theta)$
- squared error: $[y - h_\theta(x)]^2$

Stochastic Gradient Descent

- Gradient Descent (GD):

$$\theta^{(i+1)} \leftarrow \theta^{(i)} - \alpha_i \sum_n \nabla Loss\left(x_n, y_n; \theta^{(i)}\right)$$

  where $\alpha \in [0,1]$ is the step length (a.k.a. learning rate)

  n indexes data points and i indexes GD iterations

- Stochastic Gradient Descent (SGD):

$$\theta^{(n+1)} \leftarrow \theta^{(n)} - \alpha_n \nabla Loss\left(x_n, y_n; \theta^{(n)}\right)$$

  where n indexes both data points and SGD iterations

  How do we ensure convergence?

Convergence

- Robbins-Monro sufficient conditions for convergence:

  $\sum_{n=1}^{\infty} \alpha_n = \infty$ and $\sum_{n=1}^{\infty} (\alpha_n)^2 < \infty$

- Examples that satisfy Robbins-Monro sufficient conditions

  $\alpha_n = 1/n$

  $\alpha_n = 1/(\tau + n)^k$ where $\tau \geq 0$ and $k \in (0.5,1]$

- However, convergence is very slow.

AdaGrad

- Adaptive gradient
- Use a different step size for each parameter

$$\theta_m^{(n+1)} \leftarrow \theta_m^{(n)} - \frac{\alpha}{\tau + \sqrt{s_m^{(n)}}} \frac{\partial Loss\left(x_n, y_n; \boldsymbol{\theta}^{(n)}\right)}{\partial \theta_m^{(n)}}$$

where $s_m^{(n)} \leftarrow s_m^{(n-1)} + \left(\frac{\partial Loss\left(x_n, y_n; \boldsymbol{\theta}^{(n)}\right)}{\partial \theta_m^{(n)}}\right)^2$

- Often used in backpropagation

Topics Covered

- Algorithms
  - Classification
    - Nearest neighbor, mixture of Gaussians, perceptrons, neural networks, support vector machines
  - Regression
    - Linear regression, Gaussian Processes, neural networks
  - Sequence learning
    - Hidden Markov models, recurrent neural networks, recursive neural network
  - Ensemble learning
    - Bagging, boosting
- Theory and Practice
  - Overfitting, distributed learning, stream learning

Topics that we didn't cover

- Graphical Models
- Unsupervised learning
- Semi-supervised learning
- Reinforcement learning
- Active learning
- Learning theory

Other Courses Related to ML

- CS486/686: Artificial Intelligence (S17 Poupart)
- CS475/675: Computational Linear Algebra (S17)
- CS485/685: Theoretical Foundations of ML (Shai Ben-David)
- CS870: Neural Networks (S17 Jeff Orchard)
- CS898: Deep Learning and its Applications (S17 Ming Li)
- CS885: Reinforcement Learning (F17 Yaoliang Yu; S18 Poupart)
- STAT440/840: Computational Inference
- STAT441/841: Statistical Learning – Classification
- STAT442/890: Data visualization
- STAT444/844: Statistical Learning – Regression
- STAT450/850: Estimation and hypothesis testing

Master’s in Data Science

- New! Starting in Fall 2017
- Intersection of Machine Learning, Data Systems and Statistics
- https://uwaterloo.ca/data-science/