Machine Learning Crash Course

WEEK 7

Anything that could give rise to smarter-than-human intelligence - in the form of Artificial Intelligence, brain-computer interfaces, or neuroscience-based human intelligence enhancement - wins hands down beyond contest as doing the most to change the world. Nothing else is even in the same league.” ~ Eliezer Yudkowsky

Topics

- Large Margin Classification
  - Optimization Objective
  - Large Margin Intuition
  - Mathematics Behind Large Margin Classification
- Kernels
  - Kernels I
  - Kernels II

1.

Large Margin Classification

Optimization Objective

- Recall Logistic Regression & Neural Networks

$$J(\theta) = -\frac{1}{m}(\sum_{i=1}^{m} y^i log h_\theta(x^i) + (1-y^i)log(1-h_\theta(x^i))) + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

- Varying Factors:
  - Amount of data
  - Implementation
- SVNs might give a more clean way to learn the parameters

Optimization Objective <**DEMO (1)**>

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \quad z = \theta^T x \quad h_\theta(x) = \frac{1}{1+e^{-z}}$$

- Re-written Logistic Regression (Consider)
  - $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$
  - $h_\theta = 1$ to get z >> 0
  - $h_\theta = 0$ to get z << 0
- If y=1:

$$J(\theta) = -\frac{1}{m}(\sum_{i=1}^{m} log h_\theta(x^i))$$

- If z << 0: $h_\theta(x) = \frac{1}{1+e^{-\infty}} = 1 \quad J(\theta) = -\frac{1}{m}(\sum_{i=1}^{m} log 1) = 0$
- If y=0:

$$J(\theta) = -\frac{1}{m}(\sum_{i=1}^{m} log(1 - h_\theta(x^i)))$$

Support Vector Machines (2 Parts)

$$J(\theta) = \frac{1}{m}(\sum_{i=1}^{m} y^i(-log h_\theta(x^i)) + (1-y^i)(-log(1-h_\theta(x^i)))) + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

$$cost_1(\theta^T x^i) = -log h_\theta(x^i) \qquad cost_0(\theta^T x^i) = -log(1-h_\theta(x^i))$$

NEW TERMS

$$A = \frac{1}{m}(\sum_{i=1}^{m} y^i(-log h_\theta(x^i)) + (1-y^i)(-log(1-h_\theta(x^i))))$$

$$C = \frac{1}{\lambda} \qquad B = \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

$$\min_\theta C\sum_{i=1}^{m}[y^i cost_1(\theta^T x^i) + (1-y^i)cost_0(\theta^T x^i)] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

$$CA + B \qquad A + \lambda B$$

$$h_\theta(x) = \begin{cases} 1 & \theta^T x \geq 0 \\ 0 & \theta^T x < 0 \end{cases}$$

Large Margin Intuition <**DEMO (2)**>

$$\min_{\theta} C \sum_{i=1}^{m} [y^i cost_1(\theta^T x^i) + (1 - y^i) cost_0(\theta^T x^i)] + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

$$\begin{cases} \theta^T x^i \geq 1 & y^i = 1 \\ \theta^T x^i \leq -1 & y^i = 0 \end{cases}$$

- **If C is huge, we would want A = 0 to minimize the cost function**
- **How do we make A = 0?**
  - **If y = 1**
    - **A = 0 such that θTx >= 1**
  - **If y = 0**
    - **A = 0 such that θTx <= -1**

Large Margin Intuition <**DEMO (1)**>

- **Consider the black decision boundary**
  - Note the larger min difference
  - SVM chosen due the large margins between the line and the examples
- **Consider the magenta and green boundaries**
  - Note how close they are to the examples
- **Note the distance between the blue & black line: margin**
- **See the effects of C being very large**
- **See the effects of C being very small**

Mathematics Behind the Large Margin Classifier <**DEMO (1)**>

$$\vec{u}^T * \vec{v} = \vec{p} \cdot ||\vec{u}||$$

$$\vec{u}^T * \vec{v} = ||\vec{p} * \sqrt{u_1^2 + u_2^2}||$$

$$\vec{u}^T * \vec{v} = u_1 * v_1 + u_2 * v_2$$

- E.g. Let n=2
- See that:

$$\min_\theta \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

$$\min_\theta \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 = \frac{1}{2} ||\theta||^2$$

1. Kernels

Kernels I <**DEMO (3)**>

Consider more features

→ We will predict y=1

If our hypothesis ≥ 0

Consider a Gaussian kernel

Given x data points, compute the new features using "landmarks"

→ We will accomplish this using a "similarity metric"

$$f_k = similarity(x, l^k) = \exp\left(-\frac{||x - l^k||^2}{2\sigma^2}\right) = \exp\left(-\frac{\sum_{j=1}^{n}(x_j - l_j^k)^2}{2\sigma^2}\right)$$

Kernels I <**DEMO (1)**>

General Idea

→ We can learn complex decision boundaries

→ Predict (+) near landmarks

→ Predict (-) far from landmarks

→ Selecting landmarks

→ Similarity functions

Example

→ Given a training dataset S with m samples

→ Create landmarks relative to the samples

→ Whereas $f_0=1$

When we solve the following optimization problem, we get the features We do not regularize thetaθ, so it starts from 1

Kernels II <**DEMO (1)**>

Summary

→ $C = 1/\lambda$

→ Large C, gives low bias & high variance

→ Small C, gives high bias & low variance

→ Large sigma, gives high bias & low variance (more smooth features)

→ Small sigma gives low bias & high variance (less smooth features)

Thanks!

Any questions?

You can find me at:

cs.oswego.edu/~kzeller

OR

https://github.com/ECE-Engineer

Credits

- https://www.svm-tutorial.com/2014/11/svm-understanding-math-part-1/
- https://www.ritchieng.com
- https://medium.com/deep-math-machine-learning-ai/chapter-3-support-vector-machine-with-math-47d6193c82be
- https://med.nyu.edu/chibi/sites/default/files/chibi/Final.pdf