CS 581 – Advanced Artificial Intelligence

Topic: Learning – Logistic Regression

Mustafa Bilgic

http://www.cs.iit.edu/~mbilgic

https://twitter.com/bilgicm

Gradient Optimization

Motivation

- Maximize / minimize a function *f(x)*
- Typical approach
  - Take gradient of *f(x)* wrt *x* and set it to 0
  - That is, solve $\nabla f(x) = 0$
- What if there is no analytical solution to $\nabla f(x) = 0$?
- One approach is gradient ascent (for maximization of *f*) and gradient descent (for minimization of *f*)

Gradient Ascent

- Using the Taylor expansion, a function *f(x)* can be approximated at around a as
  - $f(x) \approx f(a) + \nabla f(a) * (x - a)$

Gradient Ascent

- Find maximum of $f(x)$ where there is no closed form solution
- Start with some initial guess x_0
- While change is not much
  - $x_{i+1} = xi + \eta * \nabla f(xi)$
- η is called the learning rate and it is a user specified parameter

Let's See a Simple Example

- Maximize $f(x) = -2x^2 + 8x + 10$
- Note that $\nabla f(x) = 0$ has a closed from solution for this example. We'll use this simple example just for illustration purposes

Local Optima

- Maximize $f(x) = -x^5 - 2x^4 + 13x^3 + 14x^2 - 24x$
- Note that $\nabla f(x) = 0$ has a closed from solution for this example. We'll use this simple example just for illustration purposes

Maximum Log-Likelihood Example

- $P(X = false) = \frac{e^0}{e^0 + e^w}$
- $P(X = true) = \frac{e^w}{e^0 + e^w}$
- Assume a dataset of 3 false cases and 7 true cases
- Find w that maximizes log-likelihood

How To Set The Learning Rate?

- Too big, and it overshoots
- Too small, and it takes forever
- Moreover, it should depend on the iteration

Logistic Regression

Logistic Regression

- Learns $P(Y|\boldsymbol{X})$ directly, without going through $P(\boldsymbol{X}|Y)$ and $P(Y)$
- Assumes $P(Y|\boldsymbol{X})$ follows the logistic function

$$P(Y = false \mid X_1, X_2, \cdots, X_n) = \frac{1}{1 + e^{w_0 + \sum_{i=1}^{n} w_i X_i}}$$

$$P(Y = true \mid X_1, X_2, \cdots, X_n) = \frac{e^{w_0 + \sum_{i=1}^{n} w_i X_i}}{1 + e^{w_0 + \sum_{i=1}^{n} w_i X_i}}$$

- Learning: estimate the weights $w_0, w_1, \dots, w_n$

Learning – Parameter Estimation

- Maximize (conditional) log-likelihood

$$W \leftarrow \arg\max_{W} \prod P(Y[d] \mid \boldsymbol{X}[d])$$

$$W \leftarrow \arg\max_{W} \sum \ln P(Y[d] \mid \boldsymbol{X}[d])$$

Take derivative of CLL wrt W

- See OneNote

Optimization

- No closed-form solution for W
- One solution: gradient ascent
- Good news: log-likelihood for logistic regression is concave

Regularization

- Prefer smaller weights
  - Why?

L_2 Regularization

- Objective function
  - $W \leftarrow \underset{W}{\operatorname{argmax}}\left(\sum \ln P(Y[d] \mid \boldsymbol{X}[d]) - \frac{\lambda}{2}\|W\|^2\right)$
  - Trade-off between fit to the data vs model complexity
- Assuming n features
  - $W \leftarrow \underset{W}{\operatorname{argmax}}\left(\sum \ln P(Y[d] \mid \boldsymbol{X}[d]) - \frac{\lambda}{2}\sum_{i=1}^{n} w_i^2\right)$
- Take derivate of the objective function with respect to w_i.
  - See OneNote

L_2 Regularization and Bayesian Estimation

- Unregularized version corresponds to maximum likelihood estimate of the parameters
- Bayesian means we put a prior on what we do not know. Remember $p(\theta),\ p(\theta|D), P(X|D)$.
- In this case, we put a prior on w. That is, we have a prior distribution $p(w)$.
- L_2 regularization corresponds to
  - $p(w)$ is a Gaussian distribution with zero mean and variance related to $1/\lambda$, and
  - Taking the maximum of the posterior

L_1 Regularization

- Instead of a quadratic penalty, absolute value is used
- Assuming n features
  - $W \leftarrow \underset{W}{\operatorname{argmax}}(\sum \ln P(Y[d] \mid \boldsymbol{X}[d] - \beta \sum_{i=1}^{n} |w_i|)$
- In the Bayesian case, $p(w)$ is assumed to be not a Gaussian distribution but instead a Laplace distribution

L_2 VS L_1

- L_2 forces the large weights to get closer to zero and places an emphasis on the large weights
  - Even though the weights get closer to zero, they are often not zero
- L_1 also penalizes large weights but the emphasis is not necessarily on the large weights
  - Some of the weights become zero
  - Leads to sparser representation

Categorical Features

- Logistic regression's parameters are feature weights
  - Hence, features need to have values that can be multiplied by a weight
- What if you have a binary feature?
  - Two choices: 0/1, or -1/+1.
- What if you have a categorical features that has more than two possible values, such as R, G, B?
  - Incorrect way: R=1, G=2, B=3. Why?
  - How should we handle these features?

References

- http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf
- https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression