Machine Learning Practice and Theory

Day 4 - Supervised Learning - Linear Regression

Govind Gopakumar

IIT Kanpur

Prelude

Announcements

- Programming assignment has been put up
- Feedback form is still active
- Additional Programming material will be up by weekend
- Webpage - govg.github.io/acass

Recap

Simple models for Classification

- Distance from means : Learns a line
- KNN : Learns any shape, but at high cost
- Decision Tree : Learns rectangles, but can be costly

Visualizing the boundaries

- Captures how powerful our model is
- Represents tradeoff between space/time and accuracy

Ensemble methods : Random Forests - I

Why did we need them?

- Decision trees could be hard to construct
- Structure that decision trees compute was powerful

Brief idea

- “Ensemble” models : Have multiple models
- “Subsampling” data : Don’t give all the models the same data
- “Random” features : Don’t bother with exact IG calculations!

Ensemble methods : Random Forests - II

Model overview

- Set of “trees” - hence the forest
- Set of questions / rules in a hierarchy
- Questions are weaker than earlier
- Each “tree” is given a subset of data

Benefits?

- Extremely fast in testing
- Used in real world : Kinect
- Often the go-to classifier / regressor

How to come up with a loss function

Toy setting : Find closest point

- We are given a set of N data points
- Our goal : Find the point that is closest to them all.

Casting as an optimization

- What is the appropriate loss function?
- How do we solve this problem?
- Gradient Descent??

What is the end goal?

Supervised learning

- Predict a class / value for new points
- “Train” using lots of old points, their labels
- Learn something meaningful
- Hopefully generalizes!

What's an easy method to model a trend?

Naive method of doing regression?

- Do KNN again, but choose to do regression!
- Decision Trees for regression?

What other methods exist?

- Simplest method - draw a line!

Our first regressor

Regression as line fitting

Given Input

- N examples : **training data**
- N values : **training labels**

What is the objective now?

- Given a new example : Predict what the value will be?

How do we adapt our existing model?

KNN for regression!

- Choose the values of nearby methods and average them.
- Improvement (that works for classification, but not as effective)
  - use distance to weigh them

Decision Tree for regression?

- How do we choose the split?
- How do we choose the final value to predict?

Regression as line fitting

Model overview

- Draw a line that "fits" through all the given points
- Why a line? Why not arbitrary curves?

Geometry of the problem

- There's no real decision "boundary"
- What does it look like in higher dimensions?

Linear Regression : via Optimization - I

Very toy example

- Simple 1 D regression
- Data : X (Nx1), Y (Nx1)
- Model : $y = mx$
- Geometry of this?

How do we solve the optimization problem?

- Formal objective : minimize $l(w)$
- Is there an intuitive guess?

Linear Regression : via Optimization - II

It is possible to solve this analytically!

- Let's do it without intercept
- Direct technique : Take gradient, set to zero?
- Gradient descent : Why?

With the intercept term?

- Again possible!
- Find in terms of partial derivatives!

Linear Regression : via Optimization - III

Modelling assumption

- $y = \langle w, x \rangle$
- Why does this make sense?

Can we set up a loss function now?

- What is a natural loss function?
- How do we optimize this function?

Linear Regression : via Optimization - IV

Final form of the loss function:

- $l(w) = \sum(y - \langle w, x \rangle)^2$
- Called the "squared loss"
- Obviously, other losses can be used

How do we optimize this?

- Gradient descent!
- Can we get away with a direct step?

Linear Regression : via Optimization - V

Multidimensional setting

- The same thing : $(y - \langle w, x \rangle)^2$
- Can be solved analytically!

How do we solve this?

- $\frac{\partial}{\partial w} \sum (y_i - w^T x_i)^2 = 0$
- $2(y_i - w^T x_i) \frac{\partial}{\partial w} (y_i - w^T x_i) = 0$
- Final form?
- $w = (X^T X)^{-1} X^T Y$

Linear Regression : via Optimization - VI

Mathematical issues?

- Why should the inverse exist?
- What can we say about the values of w?

Implementation issues?

- How do we invert this matrix?
- Numerical issues?

Linear Regression : via Optimization - VII

Regularizer : Why?

- Some way to control our objective function.
- What can the values of w be in our answer?
- What does it mean to have really large values?

How do we impose it?

- Requirements : restrict w somehow
- Add to the loss function?
- What does it mean intuitively?

Coming up with a MLE model?

- Consider probability of data
- Maximize this quantity

Model choices?

- How do we choose our likelihood function?
- How do we combine to find probability of data?

Linear Regression : via Probability - II

Review of Gaussian distribution

- $x \sim \mathcal{N}(\mu, \sigma^2)$
- Can model any real value
- Extends to higher dimensions as well!
- $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-1}{2\sigma^2}(x-\mu)^2}$

Why is this necessary?

- Earlier example used "Bernoulli"
- We wrote down the "probability" of our experiment
- Came to intuitive answer!

Linear Regression : via Probability - III

Model overview

- Assume points generated from a Gaussian
- $y_i \sim \mathcal{N}(w^T x_i, \sigma^2)$
- Why does this make sense?

Writing the likelihood

- $p(y_i) = ?$
- $p(Y) = ?$
- How to optimize this?

Linear Regression : via Probability - IV

Optimizing the likelihood

- Do we do gradient descent?
- How do we guarantee convexity?

Doing MLE

- We wish to maximize probability of our data being observed
- What is our final solution?

A more complicated regression problem

Matrix Factorization - I

Problem setting

- Movie Recommendations
- Item ratings

Model overview

- Assume we have a giant "matrix" of entries
- This can be "factorized" : $M = UV^T$
- If $M - NxD, U = NxK, V = DxK$

Matrix Factorization - II

Model interpretation

- $U : NxK$ what could this be?
- $V : DxK$ what could this be?
- In context of movies, what do these represent?

Model formation

- How is the rating $m_{i,j}$ formed?
- Does this make sense intuitively?

Matrix Factorization - III

How do we now solve this?

- Is there some “loss” function we can optimize?
- How do we take care of so many dependencies?

Reducing this to a known problem

- Consider a single movie and all its ratings
- How is this formed?
- Suppose someone told you the “vector” of the movie.
- How can you now solve it?

Matrix Factorization - IV

Taking a look at individual movies

- Denoted by a column, say v
- Entries in this column?
- How are they generated?

Does this relate to a known problem?

- What happens if we “know” the values of U?
- What sort of optimization technique does this generate?
- Is a solution guaranteed?

Matrix Factorization - V

Things to consider

- How did we choose k?
- What are the parameters and hyper-parameters then?
- How can we improve this?

Extensions

- Can we work with different datasets?
- Movie - user pair, as well as book - user pair?

Conclusion

Concluding Remarks

Takeaways

- How to model a trend using regression.
- Interpretation of regression as a weighted sum
- How to solve a non-trivial optimization problem
- When can an analytical solution be derived?

Announcements

- Programming tutorial
- Extra class?
- Quiz 1 (hopefully) tonight
- Open to suggestions on kinds of questions?
- Assignment 2 out by weekend

References

- Lecture 4, CS 771 IIT Kanpur
- Lecture 5, CS 771 IIT Kanpur