Machine Learning Practice and Theory

Day 7 - PCA, Kernels, Ensembles

Govind Gopakumar

IIT Kanpur

Prelude

Announcements

- Doubt clearing session on Thursday
- Programming tutorial on Gaussian Mixture Models up (or will be by tonight)
- Programming tutorial on PCA up (or will be by tonight)

Recap

Clustering

- KMeans clustering
- How to look at it as an optimization problem
- Alternating optimization

Generative modelling

- How to predict what future data looks like
- Note : Can be extended to different sorts of models

Dimensionality Reduction

Aren't more features better? - I

How many "features" should we have?

- Depends on problem
- Depends on our required model
- Depends on the data we collect

How useful are these features?

- Variance?
- Entropy?
- Mean?
- "Information Gain"?

Aren't more features better? - II

Curse of dimensionality

- We can't "fill" the space!
- What happens to our hyperplanes and other algorithms?

Feature extraction

- Some features are probably better used together!
- Some features are probably removable

Principal Components Analysis - I

Model overivew

- Find "informative" directions.
- Exclude uninformative directions.
- Flexible : choose how many you want.

What is an informative direction?

- Highest information gain?
- Do we look at combinations of features?
- How do we measure how much information we have?

Principal Components Analysis - II

Review of Covariance

- Suppose we observe two "features"
- How do we define covariance between them?
- What does this mean in machine learning?

Geometry of covariances

- How do positively correlated values look?
- How do negatively correlated values look?
- How do uncorrelated values look?

Principal Components Analysis - III

Geometry of model

- Look at data along variance
- What captures spread?
- How is it naturally low dimensions?

Computing the spread

- Denote by "covariance"
- Rank the axes in order of this "spread"
- Choose as many as you wish!

Principal Components Analysis - IV

Solving it analytically

- Let's look at the covariance of the data!
- How do we compute this?
- What can we do with this matrix?

As a linear embedding

- We wish to "embed" the points onto a line
- Choose coordinate as location along line!
- $\langle u_1, x_i \rangle$: distance along u_1

Principal Components Analysis - V

What is our loss function / optimization?

- Variance of the projections : $\frac{1}{N}\sum(\langle u_1, x_i\rangle - \langle u_1, \mu\rangle)^2$
- Simplification : $\|u_1\|_S^2$
- What is S?

Where is the optima for this?

- If we restrict u to specific vectors, we can get a solution.
- Maximum eigenvalue : variance of data
- Eigenvector : direction along this variance

Principal Components Analysis - VI

Steps in PCA

- Center the data
- Compute S matrix
- Find the eigen vectors corresponding to high eigenvalues

How many to choose?

- Flexible! Choose as much as you want.
- What do you lose?
- What do you gain?

Principal Components Analysis - VII

Why is this useful?

- Automatic feature extraction!
- Dimensionality reduction
- Quality of features

Usage

- Templates of eigenvectors
- What does this look like in practice?

Kernels

Increasing dimensionality of data - I

Wait, what?

- Why should we increase the dimensionality?
- What issue do our current methods all share?
- Geometry of the problem!

Okay, how?

- "Lift" our features into higher dimensions
- Called feature mapping / kernels
- Examples?

Increasing dimensionality of data - II

Procedure

- Come up with "mapping"
- Transform all our data using this
- Train on this data

Computational issues?

- Construction can be expensive!
- Storing these can be expensive, if we have "large" mappings.

Increasing dimensionality of data - III

Using a Kernel

- A "kernel" measures similarity in high dimensional spaces
- $k(x, z) = \langle \phi(x), \phi(z) \rangle$

Examples of kernels

- $k(x, z) = (\langle x, z \rangle)^2$
- What is the associated $\phi(x)$?
- Can we write such a mapping down for all kernels?

Increasing dimensionality of data - IV

Examples of kernels

- Quadratic kernel : $k(x, z) = (\langle x, z \rangle)^2$
- Polynomial kernel : $k(x, z) = (\langle x, z \rangle)^d$
- Radial Basis Function : $k(x, z) = \exp(-\gamma \|x - z\|^2)$

Why are they useful?

- Allow similarity in non-linear senses
- We can actually transform our data without transforming them!

Increasing dimensionality of data - V

How do we use them in models?

- Transform our "decision rule" into a dot product
- Replace dot product with kernel!

Examples

- Distance from means
- Perceptron
- Linear Regression!

Boosting and ensembles

Ensemble models - I

What?

- What is an "ensemble"?
- How do we construct it?
- Different models on same data vs same model on different data?

How?

- Aggregate or voting on predictions
- Stack : predictions as features!
- Levels of models!

Ensemble models - II

Bagging

- Create multiple copies of data
- Train similar / same models on these copies
- Aggregate predictions

Why would this work?

- Each model captures the variance of data
- Noise is spread out, reduced
- How many replications?

Ensemble models - III

Boosting

- Use only weak algorithms
- Combine them iteratively to get better predictions
- Increase weight of hard examples

Process

- Have T different "weak" models
- Start with uniform weights for all points
- Learn an initial model
- Iteratively increase weights depending on previous mistakes
- Learn better models

Ensemble models - IV

AdaBoost

- Choose a weak model (Perceptron?)
- Let it learn from data
- Check where it made errors : increase these points!
- Choose another perceptron, learn on new data

Can it learn complicated shapes?

- Yes, at times
- Outliers can screw it up a lot at times

Ensemble models - V

Comments

- Bagging vs Boosting? : no real winner
- Bagging allows parallel learning
- Boosting keeps decreasing training error

Why should we use either?

- Reduce overfitting (multiple models)
- Combine predictions

Conclusion

Concluding Remarks

Takeaways

- Dimensionality reduction techniques
- Dimensionality increasing techniques
- Why the above two are not necessarily opposites!
- Ensembling : How to convert weak predictions into strong ones

Announcements

- Doubt clearing session on Thursday : open class
- Quiz 1 is still up, please ask doubts
- Programming tutorials up for GMM, PCA

References

- Lecture 11, CS 771 IIT Kanpur
- Lecture 9, CS 771 IIT Kanpur
- Lecture 21, CS 771 IIT Kanpur