Pragmatic Programming Techniques

Sunday, November 1, 2009

Principal Component Analysis

One common problem of machine learning is the "**curse of high dimensionality**". When there are too many attributes in the input data, many of the ML algorithms will be very inefficient or some of them will even be non-performing (e.g. in nearest neighbor computation, data points in a high-dimensional space are pretty much equal distance with each other).

It is quite possible that the attributes we selected are inter-dependent on each other. If so, we may be able to extract a smaller subset of independent attributes that may still be very useful to describe the data characteristics. In other words, we may be able to reduce the number of dimensions significantly without losing much fidelity of the data.

"**Dimension Reduction**" is a technique to determine how we can reduce the number of dimensions while minimizing the loss of fidelity of data characteristics. It is typically applied during the data cleansing stage before feeding into the machine learning algorithm.

"**Feature Selection**" is a simple techniques to select a subset of features that is more significant. A very simple "filtering" approach can be used by looking at each attribute independently and rank their significance using some measurement (e.g. info gain) and throw away those that has minimum significance. A more sophisticated "wrapper" approach is to look at different subset of features to do the evaluation. There are two common model in the "wrapper" approach, "forward selection" and "backward elimination".

In **forward selection**, we start with no attribute and then start to pick the attribute with the highest statistical significance, (ie: prediction improves a lot from the cross validation check). After picking the first attribute, and we start to pair it up with another unselected attribute and find the one with the most significant improvement in the cross-validation-check. We keep growing the set of attributes until we don't find significant improvements. One issue of the "forward selection" approach is that it may miss "grouped features". For example, attribute1 and attribute2 may be insignificant when they are stand-alone, but combining them will give very big improvement.

Backward elimination can be used to handle this problem. It basically goes the opposite direction, starts with the full set of attributes and start to drop those attribute that has least statistical significance (ie: prediction degrades very little from the cross validation check). The downside of "backward elimination" is that it is much more expensive to run.

A more powerful approach called "**Feature Extraction**" is more commonly used to extract a different set of attributes by linearly combining the existing set of attributes. Principal Component Analysis "PCA" is a very popular technique in this arena. PCA can analyze the interdependency between pair of attributes and identify those significant ones.

The intuition of PCA

The intuition is to rearrange the linear combination of existing m attributes in different way to form another set of m attributes. The new set of attributes has the characteristic that

- Each attribute is independent of each other
- The set of attributes is ranked according to the range of variation

Note that attributes with narrow range of variation doesn't provide much information to describe the data samples and so can be ignored with minimal lost of fidelity. So we remove that to reduce the dimensionality.

The question is : How do we recompose the m attributes to exhibit the above 2 characteristics. Lets take a deeper look into it.

Underlying theory of PCA

Assume there are N data points in the input data set and each data point is described by M attributes. We use the statistical definition for the "mean", "variance" of each attribute and "co-variance" for every pair of attributes. Co-variance is an indicator of dependencies of two attributes

About Me

Ricky Ho

I am a software architect and consultant passionate in Distributed and parallel computing, Machine learning and Data mining, SaaS and Cloud computing.

View my complete profile

Popular Posts

MongoDb Architecture
NOSQL has become a very heated topic for large web-scale deployment where scalability and semi-structured data driven the DB requirement tow...

Designing algorithms for Map Reduce
Since the emerging of Hadoop implementation, I have been trying to morph existing algorithms from various areas into the map/reduce model. ...

NOSQL Patterns
Over the last couple years, we see an emerging data storage mechanism for storing large scale of data. These storage solution differs quite...

Couchbase Architecture
After receiving a lot of good feedback and comment on my last blog on MongoDb , I was encouraged to do another deep dive on another popular ...

Predictive Analytics: Overview and Data visualization
I plan to start a series of blog post on predictive analytics as there is an increasing demand on applying machine learning technique to ana...

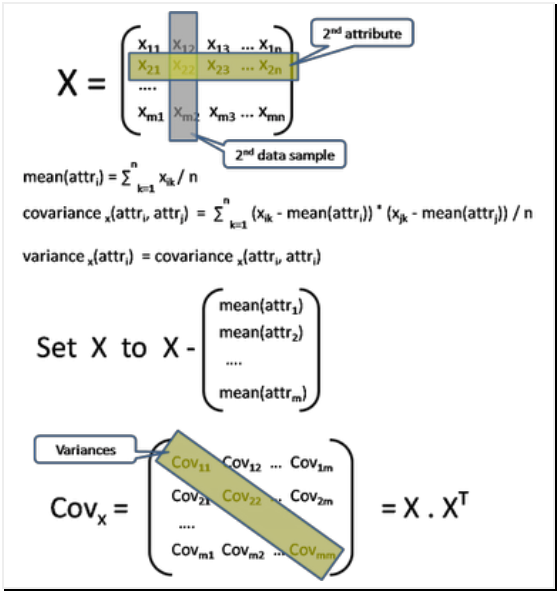
Predictive Analytics: Generalized Linear Regression
In the previous 2 posts, we have covered how to visualize input data to explore strong signals as well as how to prepare input data to a fo...

BigTable Model with Cassandra and HBase
Recently in a number of "scalability discussion meeting", I've seen the following pattern coming up repeatedly ... To make you...

Blog Archive

► 2013 (8)
► 2012 (18)
► 2011 (6)
► 2010 (18)

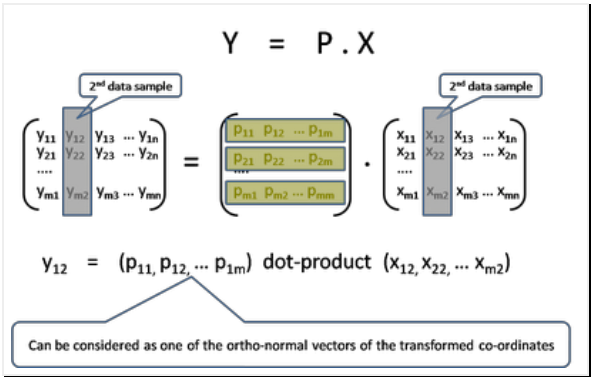
with zero implies independence.



In an ideal situation, we want COV-x to be a diagonal matrix, which means COV(i, j) to be zero. In other words, all pairs of attribute-i and attribute-j are independent to each other. We also want the diagonal to be ranked in descending order.

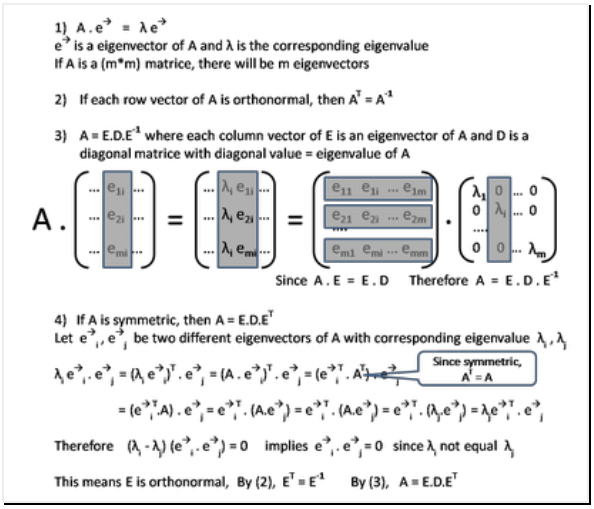
So the problem can be reduced to finding a different combination of the m attributes to form a new set of m attributes (Y = P. X) such that COV-y is a ranked diagonal matrix.

How do we determine P ?



Some Matrix theory

Here is a review of Matrice theory that will be used



Lets find the transformation matrice P

▼ 2009 (31)
► December (1)
▼ November (9)
Query Processing for NOSQL DB
Cloud Computing Patterns
Impression on Scala
NOSQL Patterns
Amazon Cloud Computing
Support Vector Machine
Machine Learning with Linear Model
What Hadoop is good at
Principal Component Analysis
► October (4)
► September (2)
► August (3)
► July (2)
► May (6)
► April (1)
► January (3)
► 2008 (22)
► 2007 (11)

Search This Blog
Search

Labels
machine learning data mining map reduce Architecture Design Cloud computing algorithm NOSQL Hadoop scalability Distributed system parallel processing big data predictive analytics Design patterns SOA performance REST ensemble method recommendation engine

Pages
• Home

$$
\begin{aligned}
Y &= P \cdot X \\
Cov_Y &= Y \cdot Y^T \\
&= (P \cdot X) \cdot (P \cdot X)^T \\
&= P \cdot X \cdot X^T \cdot P^T \\
&= P \cdot Cov_X \cdot P^T
\end{aligned}
$$

Since $Cov_X = E.D.E^T$ where E is the eigenvector of Cov_X, pick $P = E^T$

$$
\begin{aligned}
Cov_Y &= P \cdot Cov_X \cdot P^T \\
&= E^T \cdot E \cdot D \cdot E^T \cdot E \\
&= D
\end{aligned}
$$

Order the column vectors of E from left to right from the highest eigenvalue.
Now the diagonal of D is ordered by eigenvalue.

So the PCA process can be summarized in following ...

1. Input: X, a matrice of (m * n), a set of N sample data points, each with M attributes.
2. Compute Cov-X, a matrice of (m * m), the Covariance matrice of X
3. Compute the m Eigenvectors and m Eigenvalues of Cov-X
4. Order the Eigenvectors according to the Eigenvalues
5. Now found the transformation matrice P, which is a matrice of (m * m). Note that each row vector of P corresponding to an eigenvector, which is effectively the axis of the new co-ordinate system.
6. Truncate P to just take the top k rows. Now P' is a (k * m) matrice.
7. Apply P' . X to all input data to result in a matrice of (k * n). This is effectively reduce each data point from m-dimension to k-dimension.

References

A very good paper
Some Matrix math review and step by step PCA calculation

Posted by Ricky Ho at 6:28 PM

Labels: data mining, machine learning, PCA, Principal Component Analysis

2 comments:

Anonymous said...

Dear Sir,

Can you please show an working example of PCA in a step-wise (with sample data and/or free software used to achieve this), way so that I can use the same for my data.

Kindly, help me.

Thanking you,
Siva

December 3, 2009 at 10:43 PM

Jeyachandran said...

dear sir

am working PCA in image processing ,finally i found the concept of PCA but i cant understand the physical interpretation in matrix operation.

please give me some ideas about that operation ...
my mailid is
mohaideen9683@gmail.com

December 3, 2009 at 11:20 PM

Post a Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)