CS144r/244r
Network Design Project
on
Secure and Intelligent Internet of Things
(machine learning basics)
2/19/2014

Instructor: Professor HT Kung
Harvard School of Engineering and Applied Sciences
Announcements

• Today’s guest lecture: Tsung-Han Lin will give an overview of basic machine learning methods
• Dr. Robert Cohn will give a guest lecture on IoT programming next Monday
• The breadboard lab sessions went well. Many have got their micro controllers running by the end of the session
• We will have no quiz for next Monday’s class. Instead, we will be grading remaining teams on lab assignment after Monday’s class. This does not apply to those teams who have already shown their complete work in lab sessions
• It is good that project teams are being formed. Please submit to TFs members of your team by next Wed, so all students can start thinking about their projects. If help is needed, please let TFs know
• For those teams who need parts to be ordered soon, please let TFs know
• We have received 20 additional Galileo boards. Each team can have one. Please sign one out from TFs
• Various labs are hiring for summer interns. The IoT area is especially hot. Please let TFs know if you like to explore internship opportunities. Applications for most internships will need to be submitted in the next couple of weeks
What Is New in the News?

• “Whatever happened to the IPv4 address crisis?” (2/17/2014)
  • … unless everyone upgraded to IPv6, the world would be facing a crisis … for everyone. That crisis would be exacerbated by the skyrocketing demand for IP addresses due to a variety of factors: the Internet of Things (refrigerators needing their own IP address); wearables (watches and glasses demanding connectivity); BYOD (the explosion of mobile devices allowed to connect to the corporate network); and the increase in smartphone use in developing countries

• But somehow we have survived with IPv4. It will probably be the case in the foreseeable future. There are good reasons for this. It could be a good dinner discussion topic

• “More 1876 than 1995. Jim Stogdill explains how the Internet of Things is more on par with the Industrial Revolution than the digital revolution” (2/10/2014)
  • “Everyone will be affected by this collision of hardware and software, by the merging of the virtual and real”
  • “When people look back in 150 years, we think they could well say: This is when they got it. This is when they understood”

Figure 1. The prediction of IPv4 Address-Space Exhaustion (by Geoff Huston)

Corliss engine in Philadelphia’s Centennial Exposition of 1876
Recap: Motion Sensors

• Inertial sensors such as MEMS accelerometers and gyroscopes
  – Microelectromechanical systems (MEMS) contain miniaturized structures, sensors, actuators, and microelectronics
  – Google Glass uses the Invensense MPU-6050 chip which integrates a 3-axis gyroscope, 3-axis accelerometer and a Digital Motion Processor (DMP) in a small 4x4x0.9mm package

• Inertial sensors and other sensors such as image sensors and GPS play complementary roles, and can be used together (so-called sensor fusing) in consumer products
Recap: Inertial Sensors

Widely used MEMS inertial sensors

- **Accelerometers**
  - Measure *acceleration* in $x$, $y$, $z$ directions

- **Gyroscopes (or simply, gyros)**
  - Measure *angular velocity* in yaw, pitch, and roll directions

They complement each other in sensor fusion
Recap: Recognize a Gesture in a Set

- A user provides one template gesture for each of the eight vocabulary gestures offline (actually two templates are kept for each gesture; see the last slide of this lecture)
- Then online the user gives a sample gesture. The system will match the sample against the eight stored templates, and find the best match

The Nokia vocabulary of eight simple gestures:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>➔</td>
<td>□</td>
<td>➡</td>
<td>➫</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>↑</td>
<td>↓</td>
<td>❞</td>
<td>❞</td>
</tr>
</tbody>
</table>

The dot denotes the start and the arrow the end
Recap: Dynamic Time Warping (DTW)

Note that three axes: x, y and z

\[ D_0 = \text{Distance between } S[i] \text{ and } T[j] = \sqrt{(S[i].x-T[j].x)^2 + (S[i].y-T[j].y)^2 + (S[i].z-T[j].z)^2} \]

\[ d_1 = \text{DTW Distance between } S[1:i-1] \text{ and } T[1:j] \]
\[ d_2 = \text{DTW Distance between } S[1:i] \text{ and } T[1:j-1] \]
\[ d_3 = \text{DTW Distance between } S[1:i-1] \text{ and } T[1:j-1] \]

\[ D_1 = \min(d_1, d_2, d_3) \]

\[ \text{DTW distance between } S[1:i] \text{ and } T[1:j] \text{ is } D = D_0 + D_1 \]

First i items in sample time series: \( S[1], S[2], \ldots, S[i] \)

First j items in template time series: \( T[1], T[2], \ldots, T[j] \)
Today’s Topic: Machine Learning Basics

• Application examples of machine learning
• Background and challenges
• Some basic approaches in supervised and unsupervised learning
• A lot of material: We will have a session on the topic to be announced later

Readings:
– Machine Learning: A Probabilistic Perspective, Chapter 1, 2012
– Data clustering: 50 years beyond K-means, 2010
Machine Learning Today

Microsoft translates English to Chinese in real-time

Google learns face/cat detector by feeding machines tons of unlabeled images

Facebook recognizes faces and your friends
Image Classification for 1,000 Classes

Achieved accuracy: 80% within the top five selected
Why Is Machine Learning Hard?

• Too much variations in data
  – For example, in image the same object can be subject to
    • Orientation/View point
    • Pose
    • Scaling
    • Lighting
    • Occlusion
    • …
  – For some reason human brains are so good at addressing these variations

• Ultimately, we want machines to learn “ontology/concept”
What Is Machine Learning?

• More than just memorizing data
• Fundamentally, learning is an abstraction or generalization process
  – Find the underlying structure or common (invariant) features of training data
• Two broad usages
  – Prediction (continuous resolution): “How much snow are we going to have tomorrow?”
  – Classification (discrete resolution): face recognition
How to Evaluate Your Classifier?

• Classification accuracy
  – (# correct predictions) / (# testing data)

• Binary classification
  – Precision and recall
Basic Idea: Learn a Model that Maps Your Data to “Value/Label”

• For example, use a parametric model
  – Configure a black box with a set of parameters $\theta$
  – This black box will read in a data vector and output a prediction (continuous value or discrete class label)

• Formally, $y = f(x; \theta)$

Test data $x$ \rightarrow f(x; \theta) \rightarrow Predicted value/Class label $y$

• “Learning” is to determine the parameters $\theta$ using a set of training data
Learn the Right Model

• Should have some insights on why this model would suit the data well and make correct prediction/classification

• Keep the model simple (i.e., not too many parameters)
  – So your training data will be sufficient
  – So your training time will be reasonable
Roadmap for the Rest of the Lecture

• Supervised learning
  – Prediction
    • Linear regression
    • Polynomial curve fitting
  – Classification
    • Decision tree
    • Logistic regression
    • SVM

• Unsupervised learning
  – K-means
  – Gaussian mixture model
Model: straight line

\[ y = c_1 x + c_2 \]

Least squares solution!

Problem: Predict Alice’s annual income given her SAT score some years ago.
Polynomial Curve Fitting

- The “correct” model (green line) that generates data might be very complicated.
- Thus a curve fitting with a high degree polynomial would be appropriate. To specify such a polynomial, we would need sufficiently many data points.
For Noisy Data or Data with High Uncertainty, A Lot of More Data Is Need to Avoid Overfitting

You need a lot more data to learn a degree-100 polynomial
Classification

- Predicting discrete labels (the class) instead of continuous values
- Several popular models
  - Decision tree
  - Logistic regression
  - Support vector machine
Decision Trees (1/3)

- Tree structure where nodes are “questions” and the leaves determine a classification or labeling.
- We have seen these before with Xbox Kinect body post recognition.

```
Height > 65
  yes
  Weight > 150
    yes Male
    no Female
  no
  Weight < 145
    yes Female
    no Male
```
Decision Trees (2/3)

- Generally nonparametric, depending on learning algorithm. No need to define depth or number of nodes
- Goal of learning algorithm is to find features that provide the most discrimination based on class
  - For example for determining gender, asking someone’s height will yield low error rates where something like eye color gives little information
- Algorithm sketch:
  1. Find feature that provides best separation accuracy
  2. Create a node and separate data according to the feature
  3. If zero error (e.g., all examples with height > 72 are male), make a leaf node
  4. Else: recurse for each subset generated
- Common algorithm is ID3
Decision Trees (3/3)

• Advantages:
  – Model is very easy to interrogate, you can look at each node and easily understand its purpose (i.e. the feature and the threshold or value)
  – Can handle multi-domain data (real numbers, categorical data, all together)
  – Cost of classification is $O(\log n)$ where $n$ in the tree depth

• Disadvantages:
  – Optimization is most likely full of local optima and is an NP-complete problem (i.e. exact computation is too expensive)
  – Highly sensitive to the distribution of the training data (often over-fits!)
Logistic Regression

Use a linear separating plane

Probabilistic interpretation: a data point is more likely to be in one class if it is further away from the separating line
Logistic Regression: Determine the Separating Plane

- Choose the plane that maximizes the total likelihood of correct class prediction

- How to compute the probability of a data point belonging to “Male”?
  1. Project the points to normal of the separating plane
     - A large projected value means high likelihood
  2. Convert this value to a probability between [0,1]

Use logistic function for conversion

\[ f(x) = \frac{1}{1 + e^{-x}} \]
Logistic Regression: Logistic Function

Convert the projected value to probability via logistic function

\[ f(x) = \frac{1}{1+e^{-x}} \]
Logistic Regression: Summary

• What is in the model?
  – Linear separating plane
  – Logistic function for probability

• Summary
  – Training is fast
  – Very simple model with a few parameters (less chance of overfitting)
  – Prediction is also fast
  – The output of prediction has simple explanation (probability of belonging to a class)
Support Vector Machine (SVM)

- The most widely used off-the-shelf classifier
- Supports non-linear model
  - Be careful. Make sure you have enough training data and training is not too slow.
- Involves hyper-parameters
  - Means (painful) parameter tuning
- Very similar to logistic regression, but only focuses on “support vectors”
Margin

• Let us first assume data are linearly separable

• How to define the “best” separating plane?

One possibility: the one that has the biggest separation margin.

Only worry about the “support”
Most of The Time Data Are Non-separable

Include a penalty from mis-classification

Regularization parameter to weight training error

\[
\min ||w||^2 + C \sum_{i=1}^{m} \xi_i
\]

Margin

Classification error in training set
Non-linear SVM

- Map data points to a high-dimensional space and do classification there.
- Kernel trick: instead of actual mapping, define a new similarity measure between data points.
Radial Basis Function

- RBF is one of the most popular kernels
- Introduce the “variance” in the similarity measure

\[ k(x, x') = \exp\left(-\frac{(x - x')^2}{2s^2}\right) \]
How to Tune Hyper-parameters?

• Hyper-parameters describe a family of models
  – Regularization parameter C in linear SVM
  – The variance in RBF SVM

• Do not over-fit your model!
  – A model that can perfectly explain your training data is not necessarily a good model
  – Easier to over-fit when the model has more parameters

• Find the best parameter by cross-validation
Cross Validation

• Only use parts of the data for training, use the rest for testing
• Test how your model generalizes unseen training data
• N-fold Cross-validation
  – Divide your training data into N subsets
  – Use (N-1) of them to train your model, and the rest for testing
  – Repeat N times and compute average classification accuracy
## Recap: Comparison of Classifiers

<table>
<thead>
<tr>
<th></th>
<th>Decision Tree</th>
<th>Logistic Regression</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model</strong></td>
<td>Tree (horizontal/vertical separating plane)</td>
<td>Linear separating plane</td>
<td>Can also use non-linear separating plane</td>
</tr>
<tr>
<td><strong>Simplicity</strong></td>
<td>Very simple</td>
<td>Simple</td>
<td>Complicated</td>
</tr>
<tr>
<td><strong>Interpretation of output</strong></td>
<td>Clear, follow the questions asked</td>
<td>Probability</td>
<td>Unclear</td>
</tr>
<tr>
<td><strong>Chances of over-fit</strong></td>
<td>High</td>
<td>Less worry</td>
<td>Should be careful</td>
</tr>
<tr>
<td><strong>Development time</strong></td>
<td>Just plug data in</td>
<td>Just plug data in</td>
<td>Tuning tuning tuning</td>
</tr>
<tr>
<td><strong>Off-the-shelf accuracy</strong></td>
<td>Okay</td>
<td>Okay</td>
<td>Probably the highest</td>
</tr>
<tr>
<td><strong>Use case</strong></td>
<td>When features are clear</td>
<td>Often combined with neural networks</td>
<td>Often suggest RBF as a starting point if you have zero knowledge on data</td>
</tr>
</tbody>
</table>
Unsupervised Learning

• What to do when you don’t have labeled data (or when getting labeled data is difficult)?

• Automatically discover underlying data structure

• Clustering
  – K-means
  – Gaussian mixture model
K-means

- Very popular algorithms for clustering
- Minimize the sum of distances to cluster centroids

\[ J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} ||x_i - \mu_k||^2. \]
K-means: Algorithm

1. Compute the centroids of each class
2. Update membership of each example according to the closest cluster centroid
K-means May Be Stuck at Local Optimal

Centroid initialization is important
K-means

• Advantages
  – Simple, easy to implement
  – Very parallelizable
  – Easy to extend to use other distance metrics

• Disadvantages
  – Sensitive to initial approximation
  – Easily stuck at local optimal especially when dimensionality is high
  – May be difficult to determine the number of clusters
Gaussian Mixture Model

• K-means with probability
• Assume data are normally distributed around centroid
• Goal: identify the centroids, and the variance of Gaussian distributions around each centroid
Gaussian Mixture Model
Unsupervised Learning Can Help Supervise Learning

• Obtaining labeled examples can be labor intensive
• Can we teach the machines to learn some structures from unlabeled samples, and then exploit the structures for classification?
• This is semi-supervised learning
Learning Features by K-means

Training Data

Layer-1 features
(lines, dots)

Layer-2 features
(parts)

Layer-3 features
(larger parts of both faces and motorcycles)
Representation Is the Key for Machine Learning

- Representing data using the learned features

O:  X:  

Pixel 1  Eyes  Wheels

Pixel 2