• CV_M5_L01
Classifiers

CS/DSA 5970: Machine Learning Practice
Classifiers

Given some input, which of several categories does this situation belong?

• Number of categories (classes) is finite
• Used in many types of problems:
  – Is the input image an example of a cat, dog, horse?
  – Is this loan a good risk?
  – Is the tumor malignant or benign?
  – Is that a stop sign or a speed limit sign? (or others)
Classifier Formulation

• In the general case, input data can be numerical or categorical
• For our first set of examples, we will assume numerical
  – And: categorical can be transformed into numerical using One-Hot-Encoding
• We will also assume two classes for now
Classifier Formulation

• With N-dimensional numerical data, training samples are labeled points (corresponding to the classes)

• Task: identify a N-1 dimensional surface that separates the points in a way that respects the labels

• When N=2, the surface becomes a curve
  – And: the simplest (interesting) curve is a line
Drawing demo: IPAD_M5_L01b
• 2D plane with samples; separating line
• Line equation: \( f(x) = 0 \)
• Set of lines: \( f(x) = a \)
• Translating \( f(x) \) into a class label
Measuring Classifier Performance

One straw-man possibility for measuring the performance of a specific classifier: count the number of training examples that are labeled incorrectly by the current parameters.
IPAD:

• Don’t differentiate between many solutions (where some seem intuitively better than others)
• When there are errors, not clear how to change the parameters
One straw-man possibility for measuring the performance of a specific classifier: count the number of training examples that are labeled incorrectly by the current parameters

• Many solutions look the same by this metric
• For a given metric, it is not clear how to change the parameters so we can improve the classifier
A First Classifier Learning Algorithm

• Randomly choose parameters
• Measure error
• While error is too large:
  – Make small random choices to the parameters
  – If the error does not become larger, then keep the new parameters
• Done
• IPAD
A First Classifier Learning Algorithm

This is easy to implement, but:

• We could go many random steps before improving performance

• We will randomly choose a solution that minimizes cost
  – But, not all of these solutions are really the same
• IPAD
Logistic Regression

CS/DSA 5970: Machine Learning Practice
Logistic Regression

Motivation

• Want to have a smooth relationship between parameters and the cost
  – I.E., we want the cost function to be differentiable with respect to the parameters

• Want to acknowledge that examples near the dividing line are still not really acceptable
  – Instead, we want all samples far away from the dividing line
Show that $f(X)$ is a measure of distance from the line

Would like to move the line as far away as possible from the training points

But far away points should really be treated the same
Logistic Regression

Approach: add a non-linearity onto the function

- Dividing curve is still a line
- But, we can use a different cost function that is smooth in the parameter space
Logistic function

• Visualizing values on the plane, compare to linear

• Can interpret outputs as probabilities

• New cost function: squared differences

• Cost changes smoothly as we change the parameters

• Show cost as a function of one parameter
New Algorithm: Stochastic Gradient Descent

- Randomly choose parameters
- Measure error
- While error is too large:
  - For one or more training samples: compute the derivative of error with respect the parameters
  
  \[
  \frac{\partial E}{\partial w_i}
  \]
  
  For each i, compute:

  \[
  \frac{\partial E}{\partial w_i}
  \]
  
  – Change the parameters in the opposite direction
  
  \[
  w_i \leftarrow w_i - \alpha \frac{\partial E}{\partial w_i}
  \]
  
  For each i:

- Done
New Algorithm: Stochastic Gradient Descent

Notes:

• Stochastic aspect: we only compute the cost with respect to one or a small number of training samples
  – Often this is a sufficient estimate of the gradient
• Computation of the gradient is straight forward
• Depending on the training set, error may always be large
  – Change of algorithm: loop until error stops changing
• CV_M5_L03
Classes in the Infant Kinematic Data

CS/DSA 5970: Machine Learning Practice
Example: Infant Kinematic Data

Adding new columns to the infant kinematic data:

- Positions of more than just the wrists

- Assistance action type being given to the infant:
  - 0 = none
  - 1 = forward (power steering)
  - 2 = backward
  - 3 = left
  - 4 = right
  - 5 = forward (gesture)
  - 6 = backward
  - 7 = left
  - 8 = right
Preprocessing

• Compute velocity for all kinematic columns
• Drop all samples with NaNs
First Prediction Problem

Given position and velocity of all points on the body (wrists, shoulders, knees, ankles, toes): predict whether the robot is currently providing assistance

• Can be power steering or gesture-based (action type > 0)
Live demo
• CV_M5_L04
Example: First Behavior Classifier

CS/DSA 5970: Machine Learning Practice
Example: First Behavior Classifier

Stochastic Gradient Descent Classifier
• Provides a variety of linear-based classifiers
• Allows us to select from a range of different loss metrics
  – loss = ‘log’ selects logistic regression
Live demo
• CV_M5_L05
Classifier Performance Measures

CS/DSA 5970: Machine Learning Practice
Learned Model

So far:

• Model computes a score for a given input
• If the score is larger than some threshold, then we label it as being a positive example
  – For logistic regression, this default threshold is 0.5
• Contingency table: summarize correct and incorrect sorting
• Can compute other metrics: precision, recall, true positive rate, false positive rate
• Distribution of scores
• Picking a particular threshold means that the samples are sorted in some way
  – For different thresholds, we end up with different sortings & hence different metric values
  – Pierce skill score = difference between TPR and FPR:
    – Kolmogorov-Smirnov distance. Maximizes the PSS
• ROC curve
• Area under the ROC curve
• CV_M5_L06
Example: Computing Classifier Metrics

CS/DSA 5970: Machine Learning Practice
Live demo
• CV_M5_L07
Cross-Validation

CS/DSA 5970: Machine Learning Practice
Model Testing

• In large part, we do not care about the performance of a model on the data that it was trained on
  – In particular, a model can over-fit the data
• Really, we care about the performance of the model on independently drawn data
Model Testing

Ideal scenario:
• We draw some data from the world for training
• We then draw (independently) some more data from the world for testing
  – Measure performance with respect to this test data

• But: remember that model building and data sampling are stochastic processes, so performance is a random variable
  – So: we repeat the above procedure many times (at least 20-30)
Ideal Meets Reality

• In many cases, data are really expensive to collect
  – And, if the collection is inexpensive, the labeling is expensive
• Training models with more data is usually a good thing (with limits)

… can’t sample an arbitrary amount of data
K-Fold Cross-Validation (an incomplete approach)

Approach

- Cut available data into K-Folds
- Use folds 0, 1, … K-2 to train the model
- Measure performance of the model using fold K-1

- Use folds 1, 2, … K-1 to train the model
- Measure performance of the model using fold 0
- …
K-Fold Cross-Validation

Notes

• We build K different models
  – Different models do use overlapping training data

• The data used for testing a model is never used for training that model

• A data sample is used for testing exactly once
  – So, the K testing performance measures are independent of one another!
K-Fold Cross-Validation

Final note: this is only part of the Cross-Validation story

• In practice, we also want to do selection of model hyper-parameters
  – We should *never* use testing data to make these selections

• In practice, we may want to compare the performance of many different models
  – We have to tread carefully here or we can make serious statistical errors
• CV_M5_L08
Example: Cross-Validation
Live demo
• CV_M5_L09
Multi-Class Classification

CS/DSA 5970: Machine Learning Practice
Multi-Class Classification

• A linear decision surface (such as what is used in SGDClassifier) is necessarily binary

• To address multiple classes, we must construct a set of binary classifiers
  – Predictions over this set are combined together to create a single, monolithic prediction for each input
Multi-Class Classification

One-versus-one approach:

• For every pair of classes, create a classifier that distinguishes examples from the two classes

• We assume that the two classifiers randomly assign a label to all other example types (not necessarily a good assumption)

• Need $N^2$ classifiers
IPAD (continued)
Multi-Class Classification

One-versus-all approach:

• For each class, create a classifier that distinguishes examples from one class and all other classes

• Need N classifiers

• Decision surfaces can be complex, which are hard to model with a linear surface
Multi-Class Classification with the SGDClassifier

- SGDClassifier automatically detects when it is faced with a multi-class situation
- Unless forced, it will choose oneVone or oneVall, depending on the number of classes
• CV_M5_L10
Example: Multi-Class Classification
Multi-Class Classification with the SGDClassifier

Example:
• 3 classes: gesture forward, gesture left/right, all others
• Construct model, examine predictions, confusion matrix and class probabilities

Example II:
• Same, but with cross-validation
Multi-Class Classification with the SGDClassifier

Example III:

• RandomForestClassifier
Live demo
Final Notes

This particular classification problem is a challenge:

• Example uses only a small amount of data
• Labeling process leaves a lot to be desired
  – Only labeling movement as positive
  – But, one sample before the positive label will have very similar positions and velocities (and yet be labeled as negative)
  – In practice: we tend to sensor these nearby samples
Final Notes

Statistics

• We haven’t yet addressed formal methods for measuring the performance of our learned model

• One approach: with a Chi-squared test, we can formally ask whether the rows of our table are different from one-another

  – Null hypothesis: the model does not (statistically) generate a different distribution of outputs given the true class of the input

More soon…
• CV_M5_L11
Classifier Summary

CS/DSA 5970: Machine Learning Practice
Classifiers

SGDClassifier
• Numerical data
• Limited to constructing linear decision surfaces
• Must take extra steps to address multi-class cases
SGDClassifier Parameters

Some key parameters:

• Loss function
• Regularization (L1, L2 or both)
• Maximum number of iterations
• Tolerance
• Learning rate (and is it constant or adaptive)
• Early stopping (using a validation data set)
Classifiers

Looking forward to other types of classifiers:
• Non-linear decision surfaces
• Picking decision surfaces as conservatively as possible
• Allowing the algorithm to choose some training samples to ignore
• Categorical data
Classifier Metrics

- Precision & recall
- True positive rate & true negative rate
- Receiver Operator Characteristic Curve
  - Area under the ROC Curve (AUC)
- Skill scores
  - We looked at Pierce Skill Score (PSS), but there are others that address different properties
Cross-Validation

- Only report performance for data that are not used to select model parameters
- Cross-Validation explicitly does this in situations where data samples are hard to come by

More on this topic later in the semester…