E improves P on T, if the machine is learning from experience

If P (performance measure) on T does not improve, despite E, machine is NOT learning

Traditional:

Identify patterns

Write detection algorithms for each pattern (complex list of rules)

Machine Learning:

Have the machine identify unusually frequent patterns in spam flagged e-mails vs ham e-mails (the training set includes spam + ham examples)

Data mining: discover the patterns in large amounts of data, that I would not have been able to identify

Why ML?

Fine-tuning or many rules

No good solution using traditional programming (i.e., the detection of one vs two > two starts with higher pitch for SOME, but maybe not all, depending on their native language, geographical location, etc.)

Fluctuating environments > when spammers change 4U to For U

Insights > what are the patterns that it detected in the training set

Types (summary):

With or without supervision

Learn incrementally on the fly (spontaneously) or do they learn in batches

Compare new data points to training examples, or do they detect the patterns on their own (without detection algorithms written) and build a predictive model

Instance-based (learn the examples by heart, generalize based on similarity) vs model-based learning

Types

Supervised:

Solutions are labels, labelled training set

Classification or prediction (predict the target numeric value)

Or, if trained on attributes (or feature > attribute + value), and predict the probability that t belongs in a class (in the training set, you’d provide the attribute plus the feature)

1. People get e-mails every day > many > if the receiver classes one or two every day, then you take care of the training data problem > they will always have new data points according to the old class system

Unsupervised

Clustering > detect groups of similar visitors

Visualization > preserve the structure of the data as it was in the training set

Dimensionality reduction is similar to association rule learning

Anomaly detection

1. Anomaly detection > normal looks like this compared to normal do NOT look like this

Semi

Reinforcement > observe, select a policy and perform, then performance is either rewarded or penalized > update my choice of policy

Batch (training data is available, train, then if new data, delete the old data, and update the data and train again) vs online learning (incremental learning)

No longer learning during batch-learning > similar to applying a policy, where you are rewarded and penalized, but you do not update your policy

Instance-based, memorize, then use a similarity measure (count the number of words in common)

Model-based

Online

1. Systems that receive data as a continuous flow
2. Need to adapt to change rapidly or autonomously

CHALLENGE of online learning is the learning rate > Learning rate > High vs low

high > forget old spam patterns quickly (older indicators of spam)

low > less inertia, less reactive to noise

CHALLENGE > bad data can be fed into the system

Monitor input, anomalies will not be distraction, creating language p. …

Generalize to new examples, given the training, the policy you leaned, apply it to the new data, see how well you do

Instance-based, memorize spam examples, measures of similarity

Model-based, build a model based on examples, make predictions using this model

Select a model, based on plotted data, linear model, then tweak the parameters of that model, to fine-tune it

Plot the data, GDP per capita x life satisfaction

The data appears to increase linearly

Chapter 2

1. Frame the problem:

* know the business objective > is it worth investing in the area if the median housing price is low? No.
* current solution?
* Signal to noise ration must be high > improve telecommunications, for example

Labelled examples > supervised

Regression > predict a value

Univariate > single value per district

Performance measure for regression tasks > RMSE