Semantic Role Labeling and Introduction to Classification

LU XIAO LXIAO04@SYR.EDU 213 HINDS HALL

REVISED BY JENNA KIM (JKIM141@SYR.ED)

ADOPTED SOME MATERIALS DEVELOPED IN PREVIOUS COURSES BY NANCY MCCRACKEN, LIZ LIDDY AND OTHERS; AND SOME INSTRUCTOR RESOURCES FOR THE BOOK “SPEECH AND LANGUAGE PROCESSING” BY DANIEL JURAFSKY AND JAMES H. MARTIN

Attendance Checking & Download

A. Go to the Blackboard and log in with your netID
B. Content -> Week9 -> Download ‘Week9_NLP’
C. Go to the Downloads folder in your computer and unzip the file

01_Lec_9_slides_revised.pdf

02_LabWeek9.Classify.pdf

03_LabWeek9classifynames_revised.py

Outline

- Lecture
    1. Semantic Role Labeling
    2. Classification
- Break
- Lab
    1. Classification and Feature Sets

Part 1

Case Grammar
Semantic Role Labeling

Semantics of events in sentences

- In a sentence, a verb and its semantic roles form a proposition; the verb can be called the **predicate** and the roles are known as **arguments**.

When Disney offered to pay Mr. Steinberg a premium for his shares, the New York investor didn't demand the company also pay a premium to other shareholders.

Example semantic roles for the verb "pay" (using verb-specific roles)

When [$_{\text{payer}}$ Disney] offered to [$_{\text{V}}$ pay] [$_{\text{recipient}}$ Mr. Steinberg] [$_{\text{money}}$ a premium] for [$_{\text{commodity}}$ his shares], the New York investor …

CASE Grammar

- **Fillmore, Charles (1968) *"The Case for Case."***
  - A response to Chomsky's disregard for any semantics
    - "A semantically justified syntactic theory"

- Given a sentence, it is possible to say much more than this NP is the subject and this NP is the object

- Chomsky's Transformational Grammar would reduce active & passive versions of the same deep structure, but doesn't go far enough to reveal why this is possible semantically
  - *A crowbar could open that door easily.*
  - *That door could be opened easily with a crowbar.*

CASE Grammar

- Focuses on conceptual events
  - for each event or situation, there is a limited number of roles/cases which people or objects play in the situation
  - roles reflect ordinary human judgments about:
    - Who did the action?
    - Who / what was it done to?
    - What was it done with?
    - Where was it done?
    - What was the result?
    - When was it done?

Syntactic structure vs. semantic structure

- Syntactic similarities hide semantic dissimilarities
  - ***We*** *baked every Saturday morning.*
  - *The* ***pie*** *baked to a golden brown.*
  - *This* ***oven*** *bakes evenly.*
  - 3 subject NPs perform very different roles in regard to *bake*

- Syntactic dissimilarities hide semantic similarities
  - $\text{John}_{\text{agent}}$ broke the $\text{window}_{\text{theme}}$.
  - $\text{John}_{\text{agent}}$ broke the $\text{window}_{\text{theme}}$ with a $\text{rock}_{\textit{instrument}}$.
  - The $\text{rock}_{\textit{instrument}}$ broke the $\text{window}_{\text{theme}}$.
  - The $\text{window}_{\text{theme}}$ broke.
  - The $\text{window}_{\text{theme}}$ was broken by $\text{John}_{\text{agent}}$.

Cases (aka Thematic Roles or Theta Roles)

- Some of Fillmore's original set of roles still in use as general descriptors of roles
  - **Agentive (A)**
    - the instigator of the action, an animate being
      - ***John*** *opened the door.*
      - *The door was opened by* ***John****.*
  - **Instrumental (I)**
    - the thing used to perform the action, an inanimate object
      - *The* ***key*** *opened the door.*
      - *John opened the door with the* ***key****.*
  - **Locative (L)**
    - the location or spatial orientation of the state or action of the verb
      - *It's windy in* ***Chicago****.*
- Other original roles not typically used
  - **Dative (D), Neutral (N), Objective (O), Factitive (F)**

Verb-specific Roles

- Difficult to fit many verbs and roles into the general thematic roles
  - Many general sets are proposed; **not uniform agreement**
  - Generalized semantic roles now often called
    - Proto roles: Proto-agent, proto-patient, etc.
    - Or theta roles
- **Verb-specific roles** are proposed in systems
  - **PropBank** annotates the verbs of Penn Treebank
    - Extended with NomBank for nominalizations
  - **FrameNet** annotates the British National Corpus
    - Uses domains of semantically similar verbs called frames.

Propbank

- Propbank is a corpus with annotation of semantic roles, capturing the semantic role structure of each verb sense
  - Funded by ACE to Martha Palmer and Mitch Marcus at U Penn
- Each verb sense has a frameset, listing its possible semantic roles
    - Argument notation uses numbers for the annotation
    - First sense of accept (accept.01)
      - Arg0: acceptor
      - Arg1: thing accepted
      - Arg2: accepted-from
      - Arg3: attribute
- The frameset roles are standard across all syntactic realizations in the corpus of that verb sense
  - Each verb has a frameset file describing the args as above
    - Example texts are also given

Roles consistent with VerbNet

- Propbank builds on VerbNet to assign more specific roles.
- VerbNet is one extension of Levin's verb classes, giving semantic roles from about 20 possible roles
  - Agent, Patient, Theme, Experiencer, etc.
  - Similar to the theta roles
- Each class consists of a number of synonymous verbs that have the same semantic and syntactic role structure in a frame
- Whenever possible, the Propbank argument numbering is made consistent for all verbs in a VerbNet class.
  - There is only 50% overlap between Propbank and VerbNet verbs.
- **Example** from frameset file for "explore", which has a VN class:

```
<roleset id="explore.01" name="explore, discover new places or things" vncls="35.4">
<roles> <role descr="explorer" n="0">
                    <vnrole vncls="35.4" vntheta="Agent"/></role>
        <role descr="thing (place, stuff) explored" n="1">
                    <vnrole vncls="35.4" vntheta="Location"/></role>
</roles>
```

Semantic Role Notation for Propbank

- The **first two numbered arguments** correspond, approximately, to the core case roles:
  - Arg0 – Prototypical Agent
  - Arg1 – Prototypical Patient or Theme
  - Remaining numbered args are verb specific case roles, Arg2 through Arg5
- Another large groups of roles are the adjunctive roles (which can be applied to any verb) and are annotated as ArgM with a suffix:
  - ArgM-LOC – location
  - ArgM-EXT – extent
  - ArgM-DIR – direction
  - ArgM-ADV – general purpose adverbial
  - ArgM-DIS – discourse connective
  - ArgM-MOD – modal verb
  - ArgM-CAU - cause
  - ArgM-TMP - time
  - ArgM-PNC – purpose
  - ArgM-MNR - manner
  - ArgM- NEG – negation

Adjunctive and additional arguments

- **Example** of adjunctive arguments
  - Not all core arguments are required to be present
    - See Arg2 in this example.
  - Arguments can be phrases, clauses, even partial words.

When Disney offered to pay Mr. Steinberg a premium for his shares, the New York investor didn't demand the company also pay a premium to other shareholders.

Example of Propbank annotation (on demand):

[$_{\text{ArgM-TMP}}$ When Disney offered to pay Mr. Steinberg a premium for his shares], [$_{\text{Arg0}}$the New York investor] did [$_{\text{ArgM-NEG}}$ n't] [$_{\text{V}}$ demand] [$_{\text{Arg1}}$ the company also pay a premium to other shareholders].

Where for demand, Arg0 is "asker", Arg1 is "favor", Arg2 is "hearer"

Prepositional phrases and additional args

- Arguments that occur as the head of a prepositional phrase are annotated as the whole phrase
  - Consistent with other ArgM's that are prepositional phrases

    [$_{Arg1}$ Its net income] [$_{V}$ declining] [$_{ArgM\text{-}EXT}$ 42%] [$_{Arg4}$ to $121 million] [$_{ArgM\text{-}TMP}$ in the first 9 months of 1989]

- Additional arguments are
  - ArgA – causative agents
  - C-Arg* - a continuation of another arg (mostly for what is said)
  - R-Arg* - refererence to another arg (mostly for "that")

Propbank Annotations

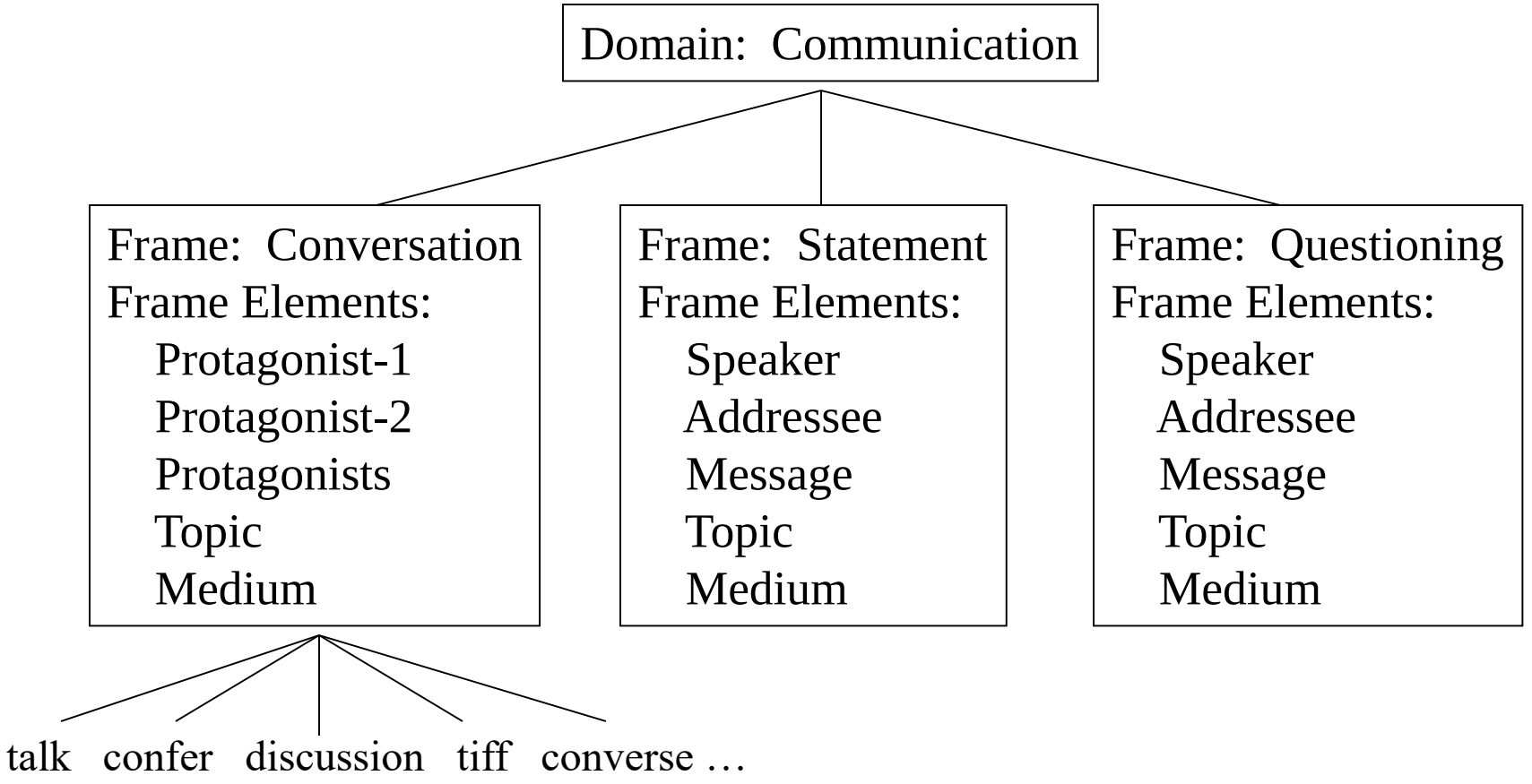
- Framesets were created by looking at sample sentences containing each verb sense.
  - ~ 4500 frames (in 3314 framesets for each verb)
- Corpus is primarily newswire text from Penn Treebank
  - Annotated the Wall Street Journal section, and, more recently, the "Brown" corpus
  - Verbs and semantic role annotations added to the parse trees
- Annotators are presented with roleset descriptions of a verb and the (gold) syntactic parses of a sentence in Treebank, and they annotate the roles of the verb.
  - Lexical sampling – annotated on a verb-by-verb basis.
  - ~40,000 sentences were annotated
- Interannotater agreement
  - Identifying argument and classifying role: 99%
    - kappa statistic of .91overall and .93 if ArgM's excluded

FrameNet

- Project at International Computer Science Institute with Charles Fillmore
  - http://framenet.icsi.berkeley.edu/
- Similar goal to document the syntactic realization of arguments of predicates in the English language
- Starts from semantic frames (e.g. Commerce) and defines frame elements (e.g. Buyer, Goods, Seller, Money)
- Annotates example sentences chosen to illustrate all possibilities
  - But recent release includes 132,968 sentences
  - British National Corpus

Example of FrameNet frames

- Semantic frames are related by topic domain



Comparison of FrameNet and Propbank

- FrameNet semantic roles are consistent for semantically related verbs (not just synonyms as in the VerbNet subset of PropBank)
- Commerce examples:

FrameNet annotation:

[$_{\text{Buyer}}$ Chuck] *bought* [$_{\text{Goods}}$ a car] [$_{\text{Seller}}$ from Jerry][$_{\text{Payment}}$ for $1000].
[$_{\text{Seller}}$ Jerry] *sold* [$_{\text{Goods}}$ a car] [$_{\text{Buyer}}$ to Chuck] [$_{\text{Payment}}$ for $1000].

Propbank annotation:

[$_{\text{Arg0}}$ Chuck] *bought* [$_{\text{Arg1}}$ a car] [$_{\text{Arg2}}$ from Jerry][$_{\text{Arg3}}$ for $1000].
[$_{\text{Arg0}}$ Jerry] *sold* [$_{\text{Arg1}}$ a car] [$_{\text{Arg2}}$ to Chuck] [$_{\text{Arg3}}$ for $1000].

Frame for buy:
Arg0: buyer
Arg1: thing bought
Arg2: seller
Arg3: price paid
Arg4: benefactive

Frame for sell:
Arg0: seller
Arg1: thing sold
Arg2: buyer
Arg3: price paid
Arg4: benefactive

Automatic SRL

- Define an **algorithm** that will process text and **recognize roles** for each verb

- Assume previous levels of Natural Language Processing (NLP) on text
  - Part-of-speech (POS) tagging,
  - Parse trees, dependency trees

- **Machine Learning classification** approaches are typical

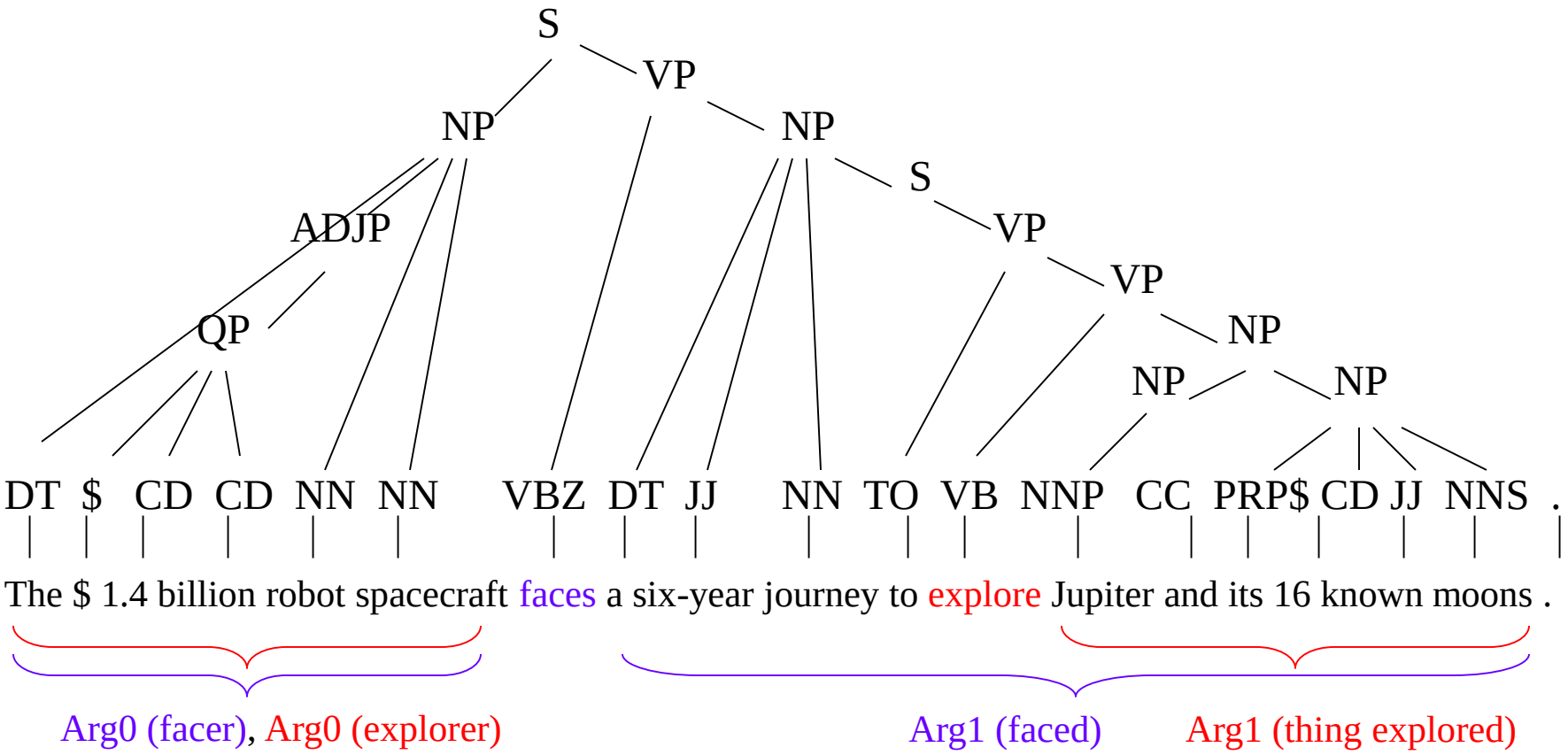
Machine Learning Approach

- Given a verb in a sentence, the problem is to find and label all arguments
- Reformulate as a classification task: For each constituent in the parse tree of the sentence, label it as to what argument, if any, it is for the verb

- For each constituent, define features of semantic roles
  - Each feature describes some aspect of a text phrase that can help determine its semantic role of a verb
    - Examples include what the verb is, POS tags, position in parse tree, etc.
- Machine Learning process:
  - Training:
    - Use annotated corpus of semantic roles with features and semantic role label
      - PropBank or FrameNet
    - ML training program uses examples to produce decision algorithm
  - Classification:
    - Run decision algorithm on text phrases and it will decide which, if any, semantic role it plays with respect to a verb

Parse Tree Constituents

- Each syntactic constituent is a candidate for labeling
- Define features from sentence processed into parse tree with Part-of-Speech tags on words



Typical Argument Features

- These features are defined for each constituent:
- PREDICATE: The predicate word from the training data.
  - "face" and "explore"
  - Usually stemmed or lemmatized
- PHRASE TYPE: The phrase label of the argument candidate.
  - Examples are NP, S, for phrases, or may be POS tag if a single word
- POSITION: Whether the argument candidate is before or after the predicate.
- VOICE: Whether the predicate is in active or passive voice.
  - Passive voice is recognized if a past participle verb is preceded by a form of the verb "be" within 3 words.
- SUBCATEGORY: The phrase labels of the children of the predicate's parent in the syntax tree.
  - subcat of "faces" is "VP -> VBZ NP"

Argument Features

- PATH: The syntactic path through the parse tree from the argument constituent to the predicate.
  - Arg0 for "faces": NP -> S -> VP -> VBZ
- HEAD WORD: The head word of the argument constituent
  - Main noun of NP (noun phrase)
  - Main preposition of PP (prepositional phrase)
- Many additional features
  - Head Word POS: The part of speech tag of the head word of the argument constituent.
  - Temporal Cue Words: Special words occurring in ArgM-TMP phrases.
  - Governing Category: The phrase label of the parent of the argument.
  - Grammatical Rule: The generalization of the subcategorization feature to show the phrase labels of the children of the node that is the lowest parent of all arguments of the predicate.

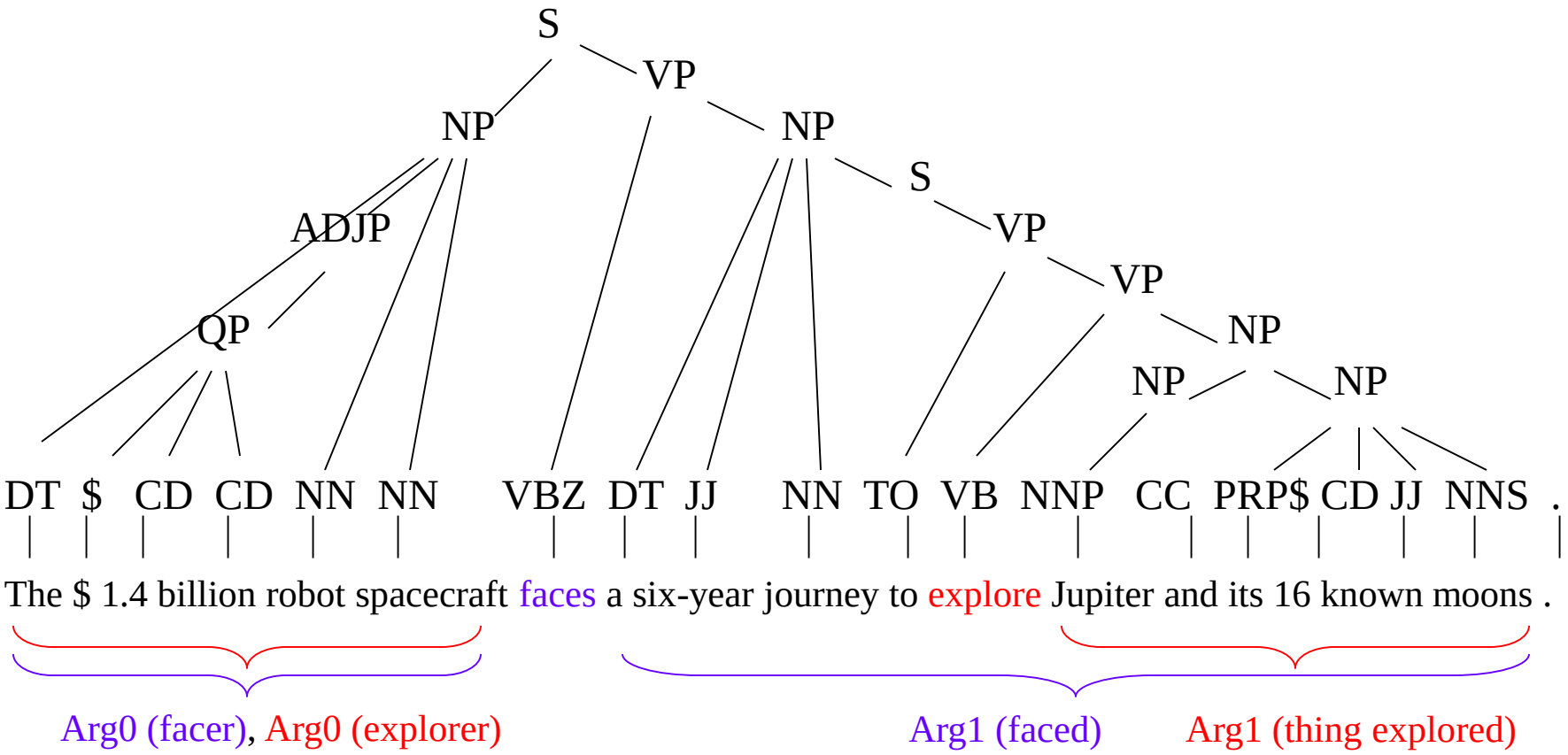
SRL problem constraints

- Results of the labeling classifier are probabilities for each label that is labels that constituent

- Use these with constraints to assign a label
  - Two constituents cannot have the same argument label,
  - A constituent cannot have more than one label
  - If two constituents have (different) labels, they cannot have any overlap,
  - No argument can overlap the predicate.

Parse Tree Constituents

- Each syntactic constituent is a candidate for labeling
- Define features from sentence processed into parse tree with Part-of-Speech tags on words



Difficulties for classification

- For each verb in a sentence, the number of constituents in the parse tree are large compared to the number of semantic roles
  - Can be hundreds of constituents eligible to be labeled a role
  - Leads to the problem of too many “negative” examples
- What should the features be?
  - Words are typically the features for an NLP problem
  - Need more about the syntactic structure as well as other potential clues
  - Typical number of features can be up to 20,000, requiring a classification algorithm that is robust for large numbers of features

State-of-the-Art on Semantic Role Labeling

Marcheggiani, D., & Titov, I. (2017). Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*.

A version of graph convolutional networks (GCNs), a recent class of neural networks operating on graphs, over syntactic dependency trees are used as sentence encoders, producing latent feature representations of words in a sentence. The stacked GCN and LSTM layers produce the best reported scored on the standard benchmark (CoNLL-2009) both for Chinese and English.

$\Rightarrow$ **Many NLP tasks such as SRL can be solved using machine learning techniques**

Part 2

Introduction To Classification:

An Example Of Supervised Machine Learning

Machine Learning

Categories: types of Learning

- Supervised : ***Classification***, Regression
- Unsupervised : Clustering
- Reinforcement : Neural Networks

Key Points

3 Important Concepts:

A. Classification

B. Training and Test Set

C. Evaluation Metrics (Precision, Recall, F-Measure)

Key Points

A. *Classification*

Let's watch the first video that introduces basic concepts of machine learning.

(source: video tutorials taught by Luis Serrano on Youtube)

1. Introduction to classification
https://www.youtube.com/watch?v=IpGxLWOIZy4

Break

Key Points

B. Training and Test Set

C. Evaluation Metrics (Precision, Recall, F-Measure)

Let's watch the second video.

(source: video tutorials taught by Luis Serrano on Youtube)

2. Testing and Evaluation Metrics

https://www.youtube.com/watch?v=aDW44NPhNw0

Classification: Definition

- Given a collection of examples (*training set*)
  - Each example is represented by a set of *features*, sometimes called *attributes*
  - Each example is to be given a label or class
- Find a *model* for the label as a function of the values of features.
- Goal: previously unseen examples should be assigned a label as accurately as possible.
- A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Supervised Vs. Unsupervised Learning

- Supervised learning (**classification and other tasks)**
  - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
  - New data is classified based on the training set
- Unsupervised learning (includes **clustering)**
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

NLP Tasks

- Many NLP tasks can be accomplished either through
  - unsupervised techniques, sometimes also called rule-based or symbolic techniques
  - Supervised techniques, where the task is defined automatically from a training set

- In both cases, the evaluation of the task will most likely use a **training set** to define the technique and a **test set** for evaluation
  - POS tagging uses Hidden Markov Models
  - Parsing uses statistical lexicalized parsers
  - **Sentiment analysis uses classification**

- The evaluation of these tasks often uses ideas from the evaluation of classification

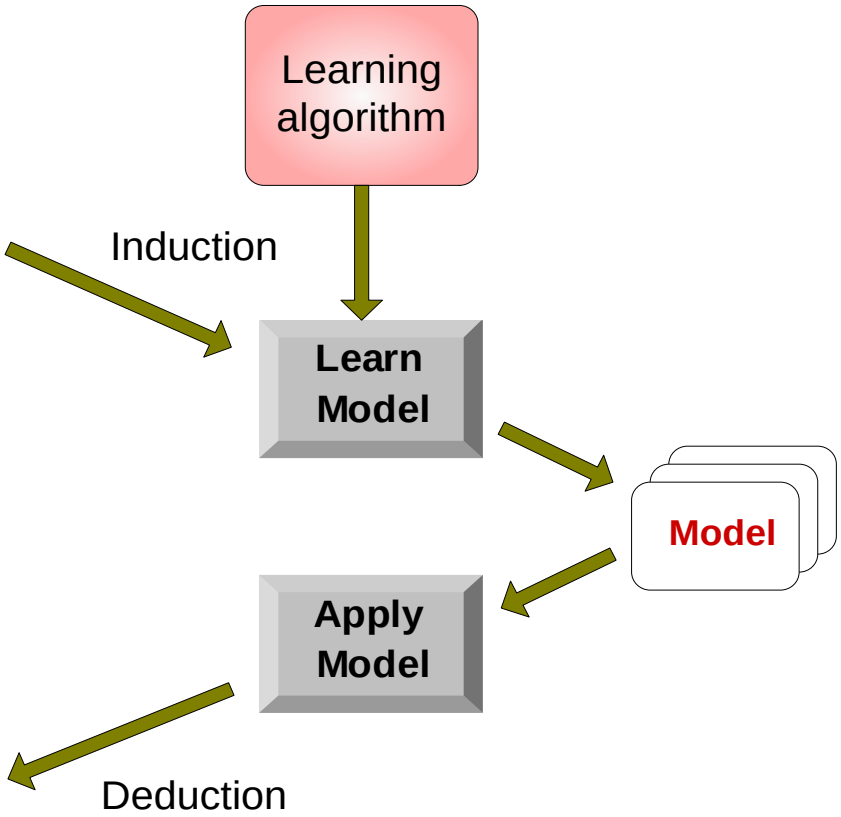
Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Classification Techniques

- There are a number of different ***classification algorithms*** to build a model for classification
  - **Decision Tree based Methods**
  - **Neural Networks**
  - **Naïve Bayes and Bayesian Belief Networks**
  - **Support Vector Machines**

- In this introduction, we illustrate classification tasks using Decision Tree methods

- Features can have **numeric values (continuous)** or a **finite set of values (categorical/nominal)**, including **boolean true/false**

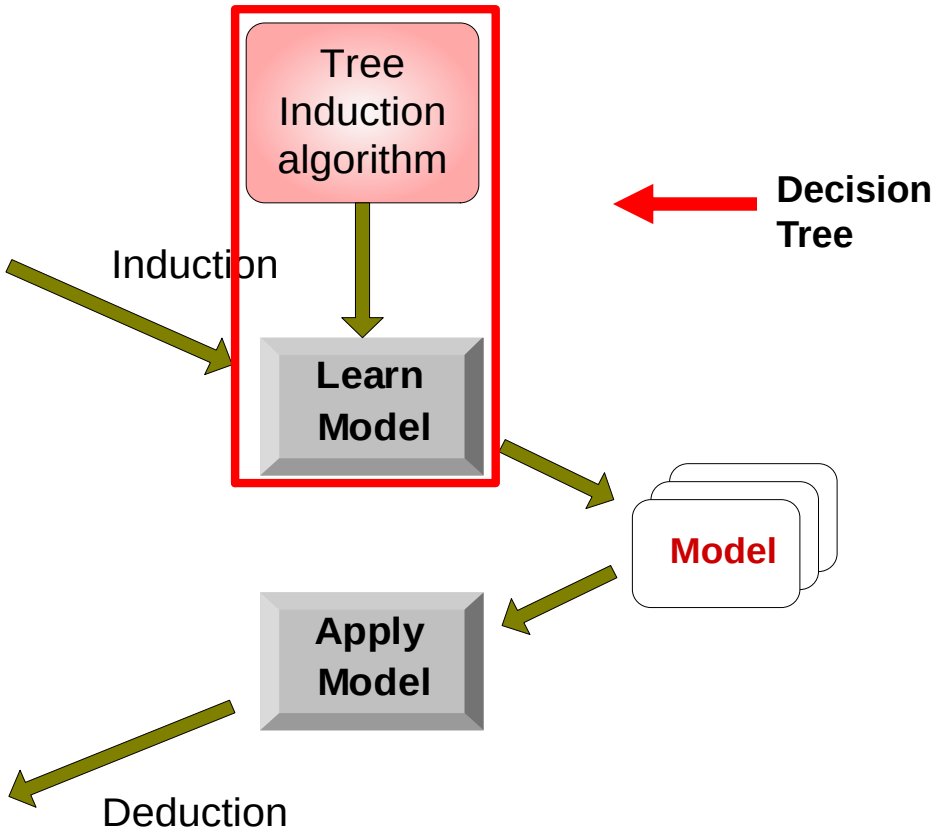
Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

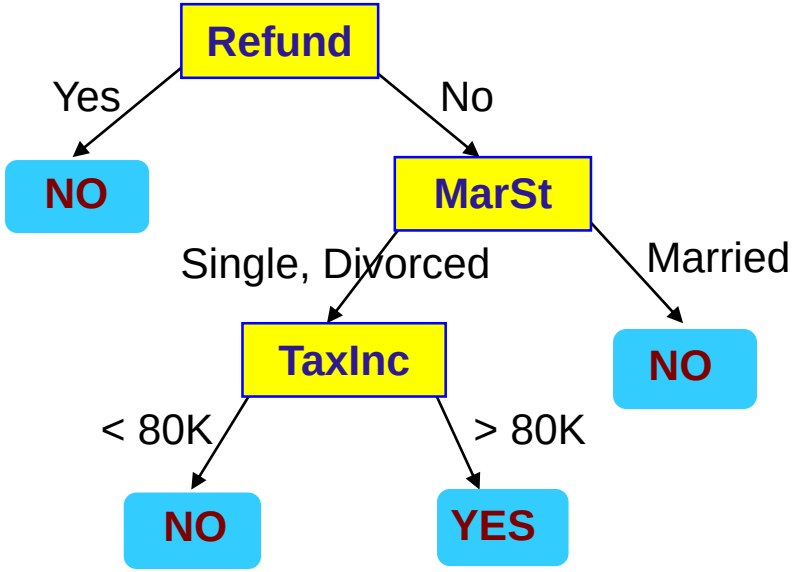
Test Set



Example Of A Decision Tree

boolean categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Training Data

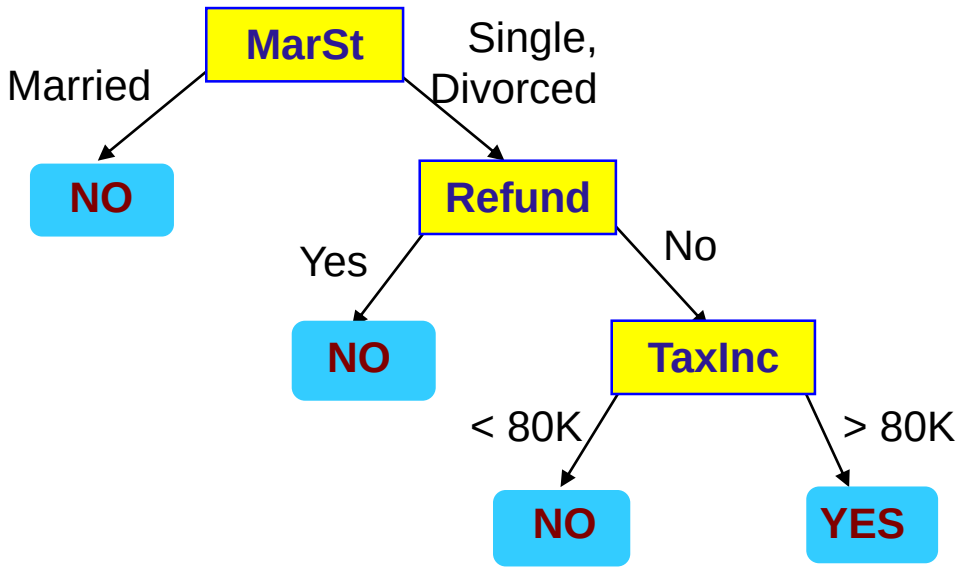
Model: Decision Tree

Example task: Given the refund status, marital status, and taxable income of a person, label them as to whether they will cheat on their income tax.

Another Example Of Decision Tree

boolean categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



There could be more than one tree that fits the same data!

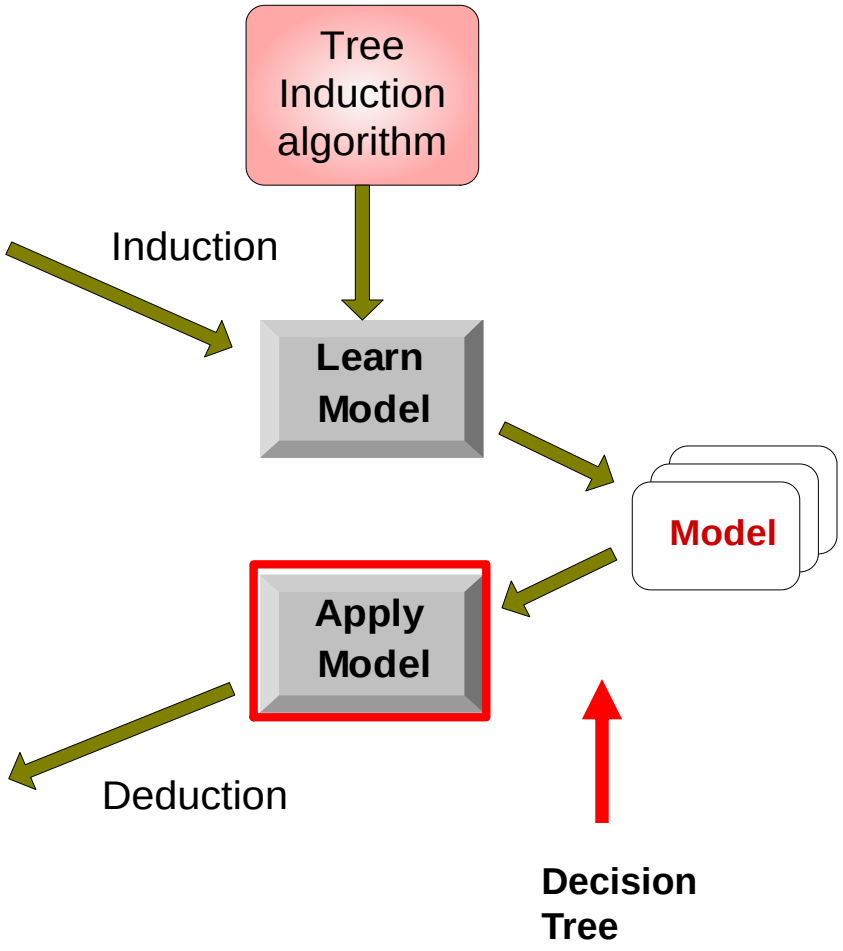
Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

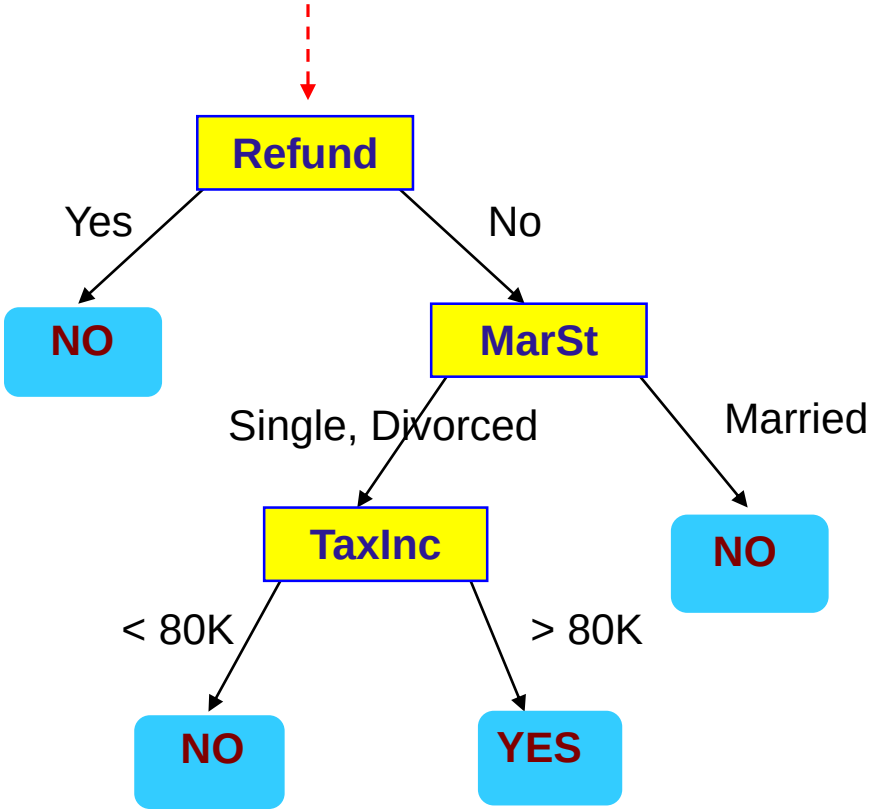
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Apply Model To Test Data

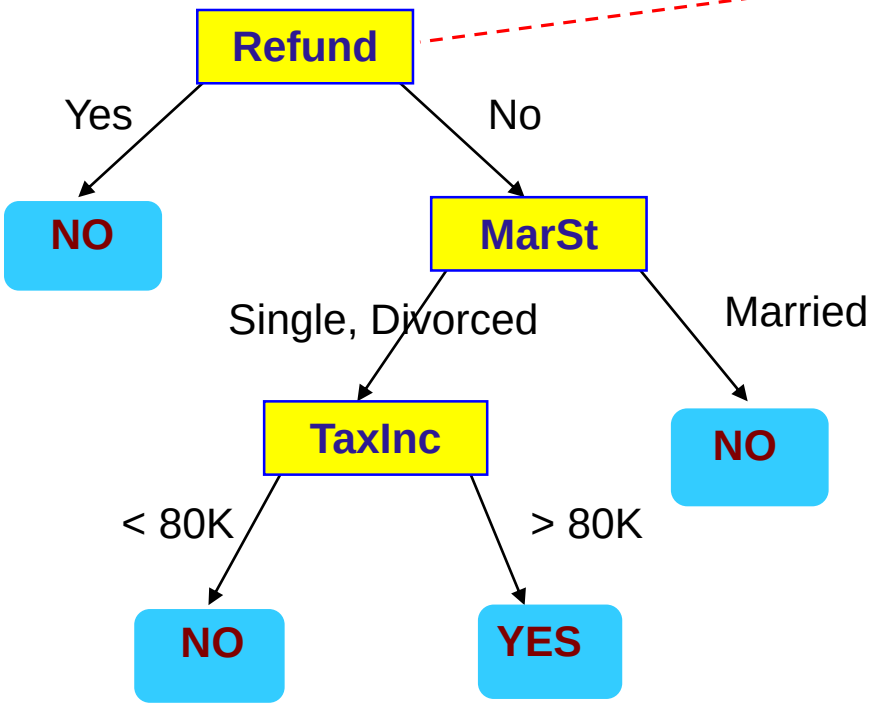
Start from the root of tree.



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

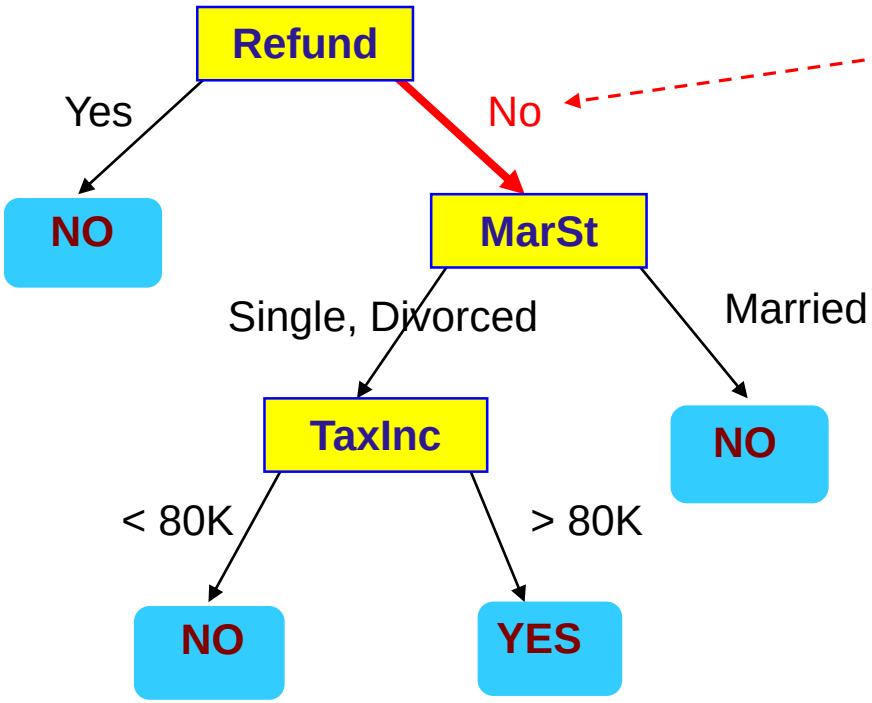
Apply Model To Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model To Test Data



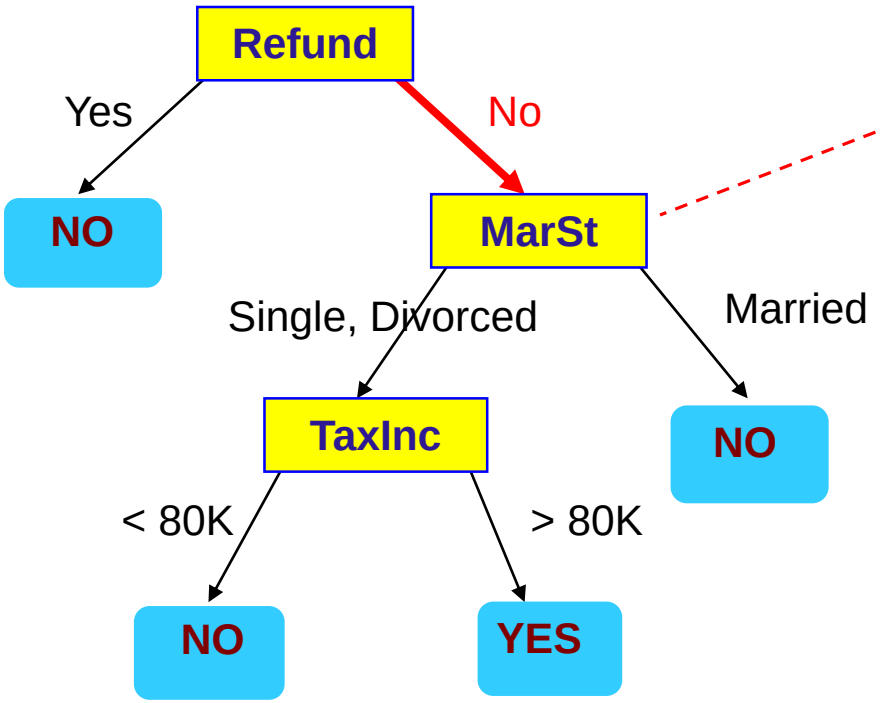
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model To Test Data

Test Data

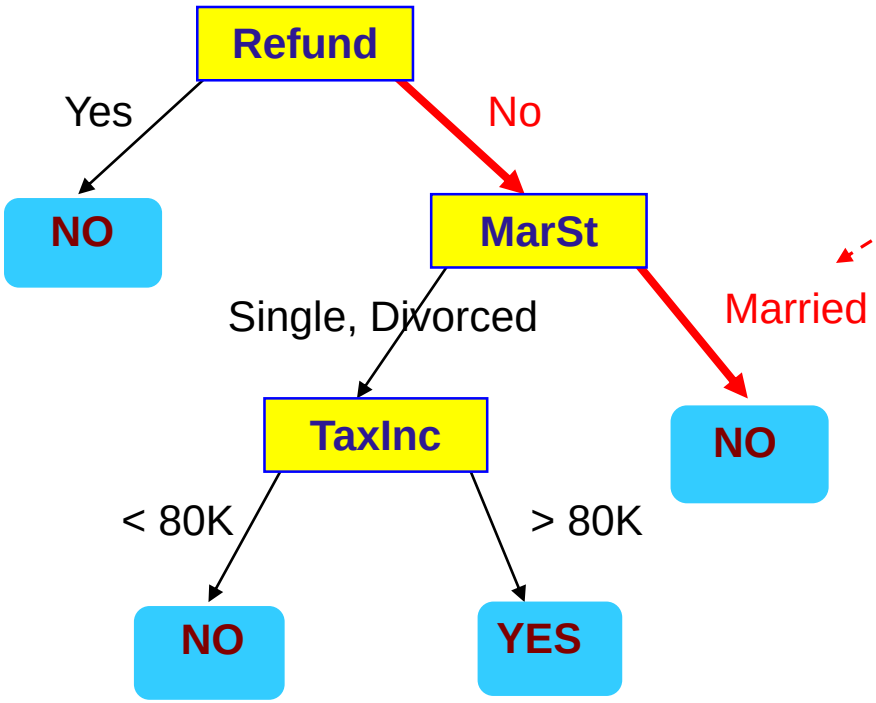
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model To Test Data

Test Data

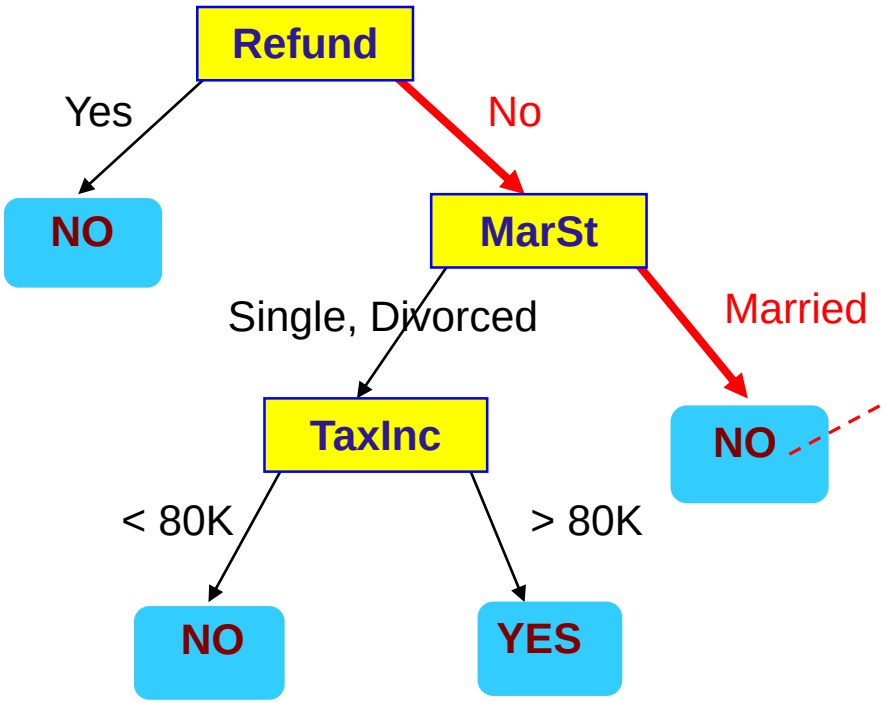
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model To Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to “No”

Performance Evaluation

Q: How good is a model?

=> Performance Evaluation

Metrics For Performance Evaluation

- Focus on the **predictive capability of a model**
  - Rather than how fast it takes to classify or build models, scalability, etc.

- **Confusion Matrix** for a binary classifier (two labels) on test set:

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Classifier Evaluation Measures

- **Accuracy** of a classifier M is the percentage of test set that are correctly classified by the model M

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

	Predicted Yes - C_1	Predicted No - C_2
Yes - C_1	**a: True positive**	b: False negative
No - C_2	c: False positive	**d: True negative**

Precision = TP/(TP+ FP),
percent correct out of all predicted Yes

Recall = TP/(TP + FN),
percent correct out of all actual Yes

F-Measure = 2 * (Recall * Precision) / (Recall + Precision)

Multi-Class Classification

- Most classification algorithms solve **binary classification tasks**, while many tasks are naturally multi-class, i.e. there are more than 2 labels
- Multi-Class problems are **solved by training a number of binary classifiers** and combining them to get a multi-class result
- Confusion matrix is extended to the multi-class case
- Accuracy definition is naturally extended to the multi-class case
- Precision and recall are defined for the binary classifiers trained for each label

Issues with imbalanced classes

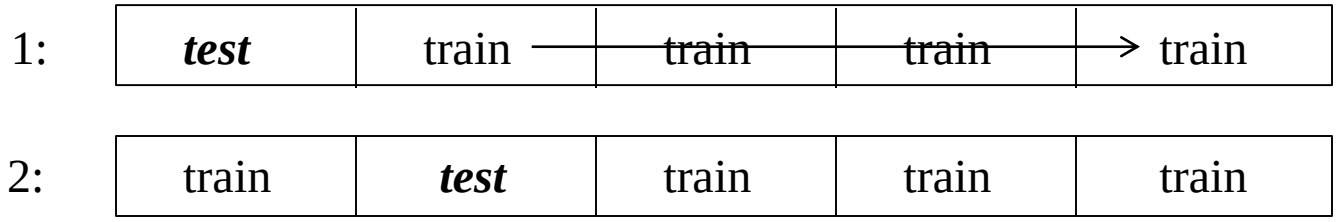
- Consider a 2-class problem with labels Yes and No
  - Number of No examples = 990
  - Number of Yes examples = 10

- If model predicts everything to be No, accuracy is 990/1000 = 99 %
  - **Accuracy is misleading** because model does not detect any Yes example
  - **Precision** and **recall** will be better measures if you are training a classifier to find rare examples.

Evaluating the Accuracy of a Classifier

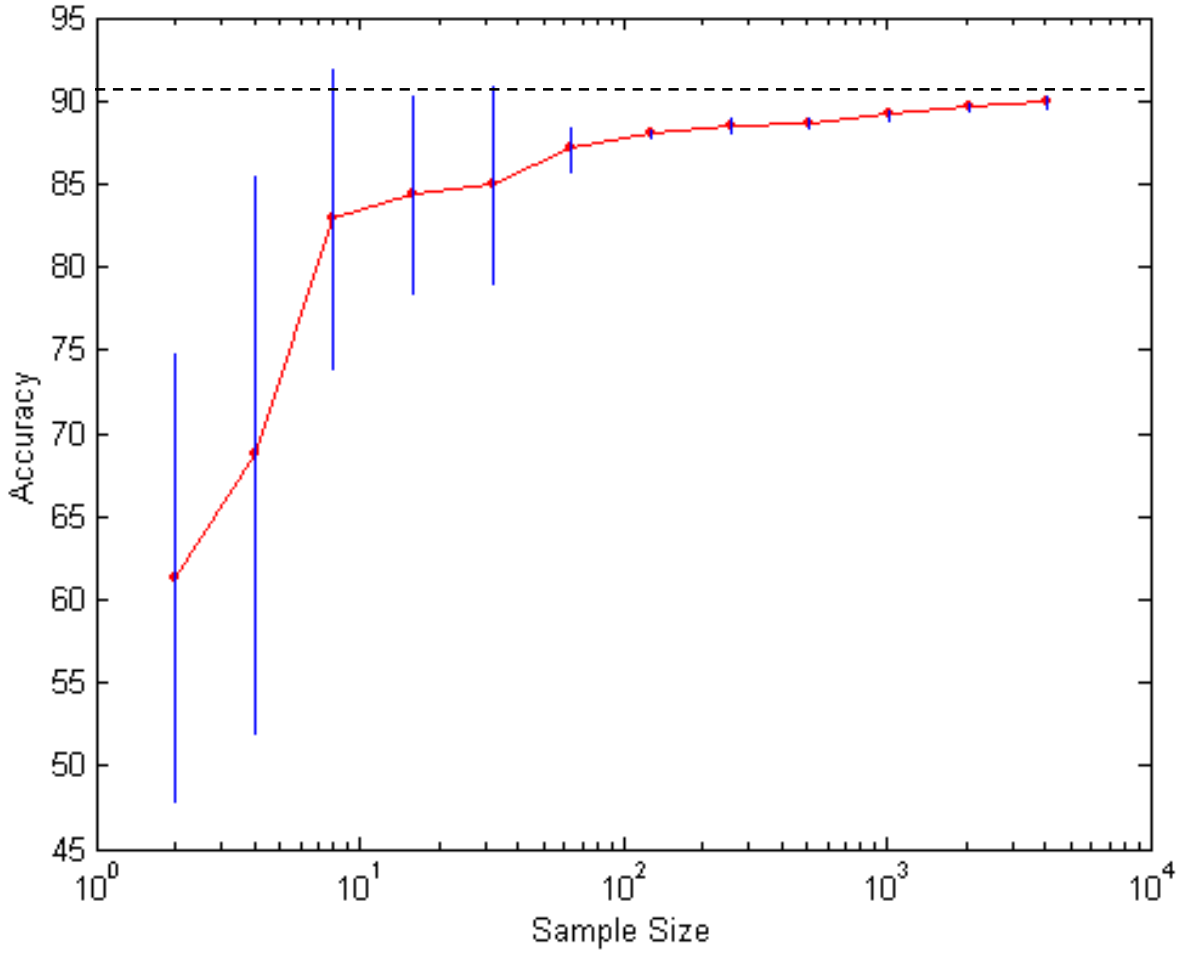
- Holdout method
  - Given data is randomly partitioned into **two independent sets**
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation

- Cross-validation (*k*-fold, where k = 10 is most popular)
  - Randomly partition the data into ***k mutually exclusive*** **subsets**, each approximately equal size
  - At *i*-th iteration, use D_i as test set and others as training set



. . .

Evaluating the Model - Learning Curve



- Learning curve shows how accuracy changes with varying sample size
- Requires a sampling schedule for creating learning curve

Classifier Performance: Feature Selection

- Too long a training or testing is a performance issue for classification of problems with **large numbers of attributes** or **nominal attributes with large numbers of values**

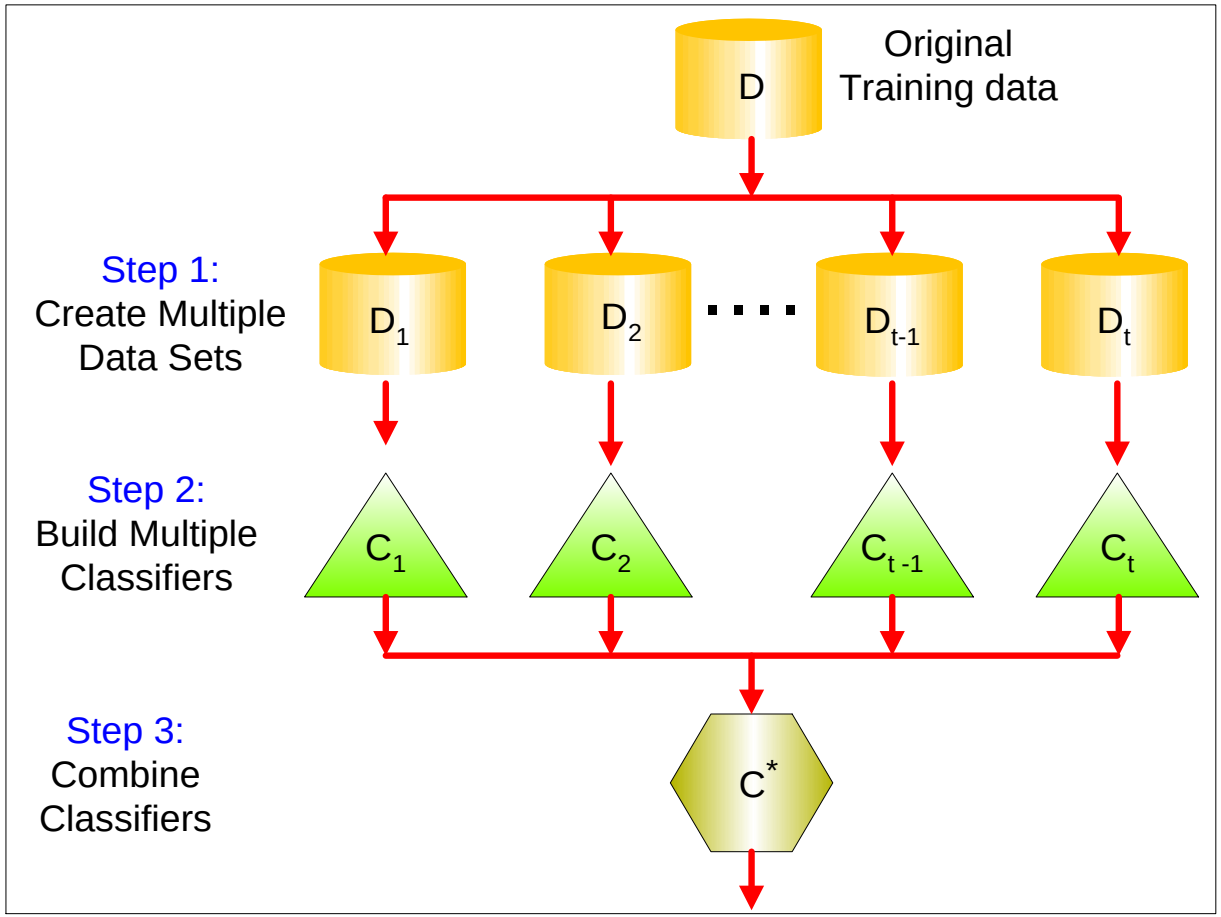
- **Feature selection techniques** aim to **reduce the number of features** by finding a smaller or minimal set that can accurately classify the problem
  - reduce the training and prediction time by eliminating noisy or redundant features

- **Two main types of techniques**
  - **Filtering methods** apply a statistical or other information measure to the attribute values without running any training and testing
  - **Wrapper methods** try different combinations of attributes, run cross-validation evaluations and compare the results

Classifier Performance: Ensemble Methods

- **Construct a set of classifiers** from the training data
- Predict class label of previously unseen records by **aggregating predictions made by multiple classifiers**
- Examples of ensemble methods
  - Bagging
  - Boosting
  - Heterogeneous classifiers trained on different feature subsets
    - sometimes called mixture of experts

General Idea: Ensemble Methods



Examples of Classification Problems

- Some **NLP** problems are widely investigated as **supervised classification** problems, and use a variety of problem instances

  - **Text categorization**: assigning topic labels to documents
  - **Word Sense Disambiguation**: assigning a sense to a word, as it occurs in a document
  - **Semantic Role Labeling**: assigning semantic roles to phrases in a sentence

- From the NLTK book, chapter 6:
  - ***Classify first names according to gender -> lab session***
  - Document classification (text categorization)
  - Part-Of-Speech tagging
  - Sentence Segmentation
  - Identifying Dialog Act types
  - Recognizing Textual Entailment

Example: Text Categorization

- Represent each document by the words/tokens/terms it contains
  - Sometimes called unigrams, sometimes bag-of-words
- Identify terms from the document text
  - Remove symbols with little meaning
  - Remove words with little meaning – the stop words
  - Stem the meaningful words
    - Remove endings to get root of the word
      - From *enchanted, enchants, enchantment, enchanting*, get the root word *enchant*
  - Group together words into phrases (optional)
    - Proper names or other words that are likely to have a different meaning as a phrase than the individual words
  - After grouping, may also want to lowercase the terms

Document Features

- Use a **feature vector** to represent all the words in a document – **one position for each word** in the collection, representing the **weights (often frequency)** of words
  - *"Water, water everywhere, and not a drop to drink!"*

$$(\underset{\text{water}}{2},\ \underset{\text{everywhere}}{1},\ \underset{\text{not}}{1},\ \underset{\text{drop}}{1},\ \underset{\text{drink}}{1}, 0, \ldots)$$ (shown with frequency weights)

- Another document with the word drink:
  - *"drink …"*

$$(\underset{\text{water}}{0},\ \underset{\text{everywhere}}{0},\ \underset{\text{not}}{0},\ \underset{\text{drop}}{0},\ \underset{\text{drink}}{1}, 0, \ldots)$$

- Feature vectors may have thousands of words and are often restricted by a threshold frequency of 5 or more

Break before Lab Session:
10 minutes