Input Representations

Machine Learning

Introduction I

Machine Learning (ML) algorithms take a set of data points as input and computes relationships or find structure in the data.

Supervised Machine Learning: Given a set of data points $\{\mathbf{x}_i, y_i\}$ $(i = 1 \ldots n)$, where y_i can be a finite set of (class) labels or real values, what is the relationship between $\mathbf{x}$ and y?

Unsupervised Machine Learning: Given a set of data points $\mathbf{x}_i$ $(i = 1 \ldots n)$, what structure exists in $\mathbf{x}$?

There are two other kinds of machine learning: *semi-supervised learning* and *reinforcement learning*. In this course, our focus would mostly be on supervised and unsupervised learning only.

Introduction II

In these slides, we will answer the following questions regarding the inputs to a machine learning algorithm:

- What is $\mathbf{x}$?
- What does $\mathbf{x}_i$ represent?
- How do we get $\mathbf{x}$?

A simple case of Iris Flower Classification I

This is a classical example[1] in ML. The goal is to classify to which class a flower (Iris flower image) belongs.

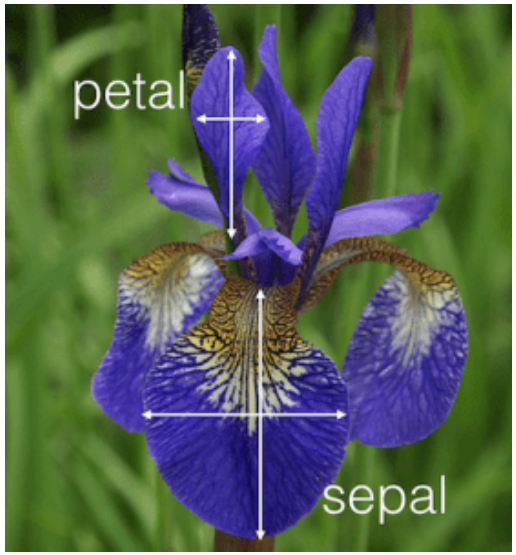

Iris Virginica


Iris Setosa


Iris Versicolor

What feature should we use for this classification problem?

A simple case of Iris Flower Classification II



After looking at the flowers, it seems reasonable to use (1) sepal length in cm, (2) sepal width in cm, (3) petal length in cm, (4) petal width in cm.

We can use these four attributes: (x_1, x_2, x_3, x_4).

A simple case of Iris Flower Classification III

x_1	x_2	x_3	x_4	y
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
...	...	...	...	...
7.0	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
...	...	...	...	...
6.3	3.3	6.0	2.5	Iris-virginica
5.8	2.7	5.1	1.9	Iris-virginica

Here is the sample dataset (50 of each class; total 150 instances)

A simple case of Iris Flower Classification IV

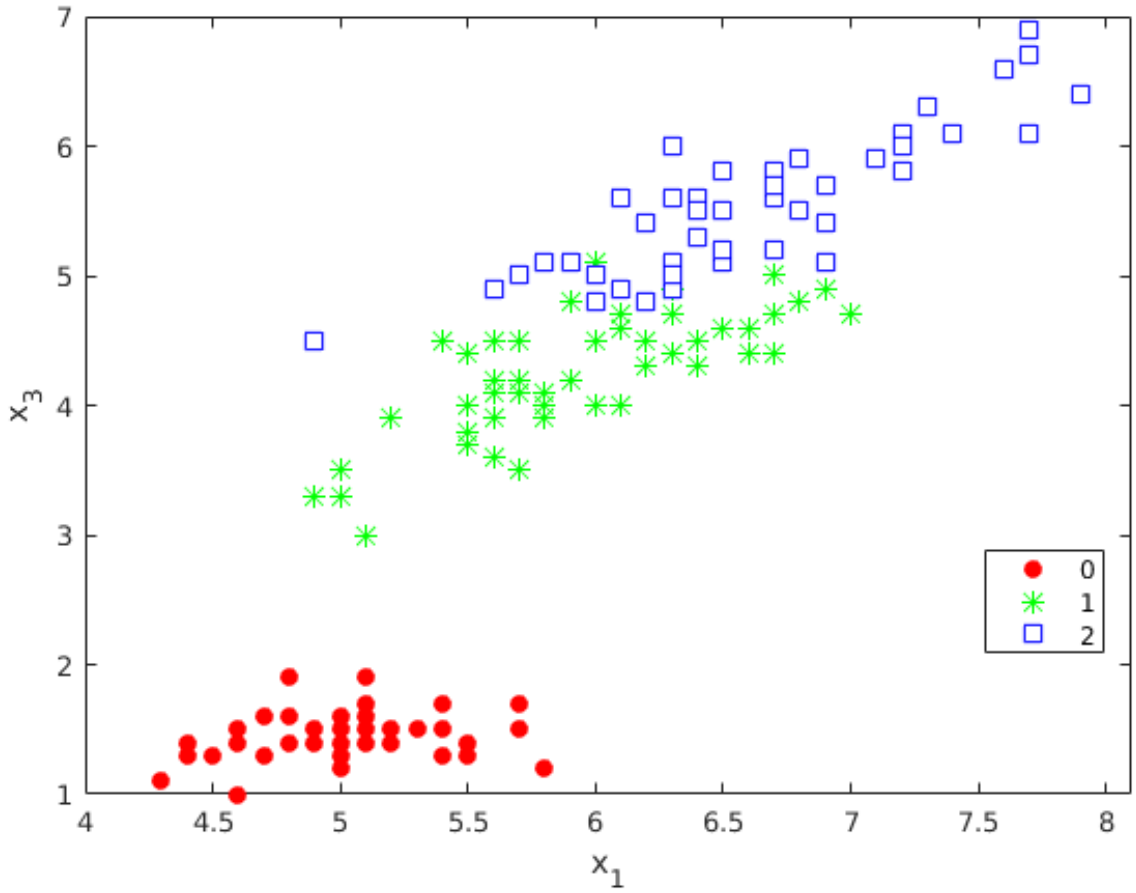


Figure: x_1 and x_3 may be enough for decision making

[1] https://archive.ics.uci.edu/ml/datasets/iris

Representation I

Representation: The attributes used to describe examples are intrinsic properties of the examples that are believed to be important.

Examples:

Task	**Attributes**
Chess	Position of white king
Drug effectiveness	Molecular structure
Class of flowers	Petal and sepal length and width

However, deciding on which attributes are important for a particular task is upto the expert(s): **you**.

Representation II

Now, we look at different tasks for which we have to decide on a input representation most suitable. For example:

Task	**Inputs**
Spam e-mail classification	Texts
Image classification	Images
Image annotation	Images
Time-series (Reg & Clf)	Real numbers
Audio classification	Audio signal
Learning Social networks	Graphs
Video classification	Videos
Predicting drug effectiveness	Molecules

Text I

Let's look at a well-known example dataset fo :

```
0    ham  Go until jurong point, crazy.. Available only ...
1    ham                      Ok lar... Joking wif u oni...
2   spam  Free entry in 2 a wkly comp to win FA Cup fina...
3    ham  U dun say so early hor... U c already then say...
4    ham  Nah I don't think he goes to usf, he lives aro...
```

Any machine learning tool can not directly take this dataset for processing. It needs **x**: Inputs and y: output $= \{\text{ham, spam}\}$.

We will extract **x** as a set of attributes.

Two approaches:

- Bag of words (BoW)
- Term Frequency-Inverse Document Frequency (TF-IDF)

Text II

BoW representation:

- Generates feature-vector by counting occurence of each word from the domain of words (Domain: Set of all words appearing in the data after preprocessing*).

*Preprocessing: (1) converting all words (except, names) to small letter, (2) removing stop words

E.g. Let say, we have two documents (or emails, or sentences):

(1) John likes to watch movies. Mary likes movies too.
(2) John also likes to watch football games.

Then, we construct a list of words for both:
"John", "likes", "to", "watch", "movies", "Mary", "likes", "movies", "too"
"John", "also", "likes", "to", "watch", "football", "games"

Text III

Now, we construct the BoW for both:
BoW1 =
{"John":1, "likes":2, "to":1, "watch":1, "movies":2, "Mary":1, "too":1
}
BoW2 =
{"John":1, "also":1, "likes":1, "to":1, "watch":1, "football":1, "games":1}

We can now represent each document against the domain of words in both the documents. This will form equal length vectors.

Here, domain is: {John, likes, to, watch, movies, Mary, too, also, football, games}

Doc1: [1, 2, 1, 1, 2, 1, 1, 0, 0, 0]
Doc2: [1, 1, 1, 1, 0, 0, 0, 1, 1, 1]

Text IV

- These numeric representation is called Term frequency representation.
- Let's write each TF as: $tf(t, d) = f_{t,d}$, where t is the term, and d is the documentID.
- Other representations are:
  - Boolean "frequencies": $tf(t, d) = 1$ if t occurs in d and 0 otherwise
  - Term frequency adjusted for document length: $f_{t,d}/(\text{number of words in } d)$
  - Logarithmically scaled frequency: $tf(t, d) = \log(1 + f_{t,d})$

Inverse document frequency (IDF):

- Measure of how much information the word provides, i.e., if it's common or rare across all documents (D).
- Mathematically,

$$idf(t, D) = \log \frac{|D|}{|d \in D : t \in d|}$$

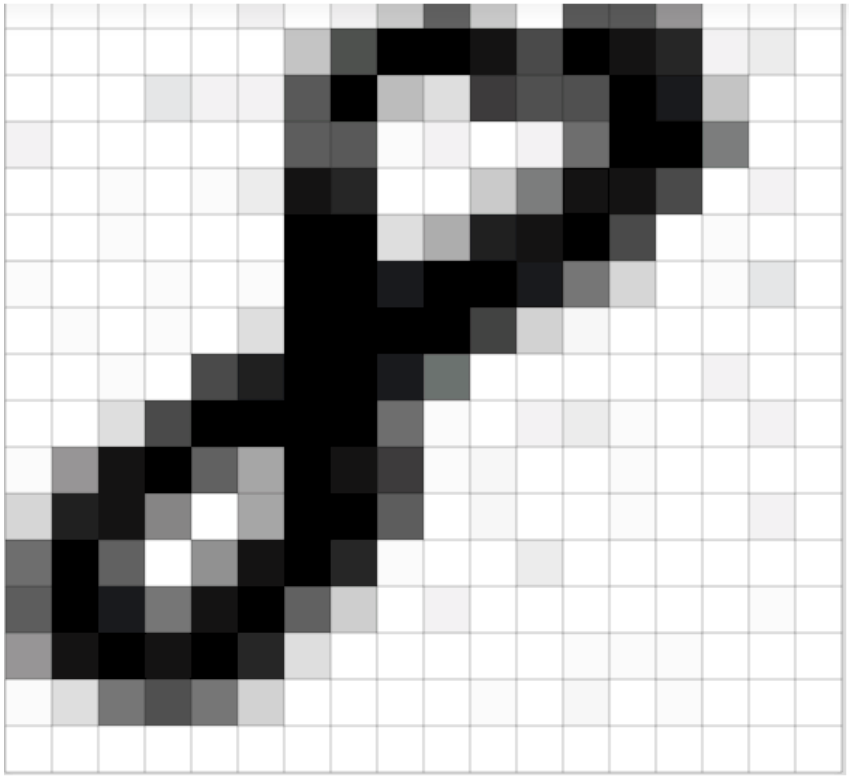
TF-IDF representation:

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$

Images I

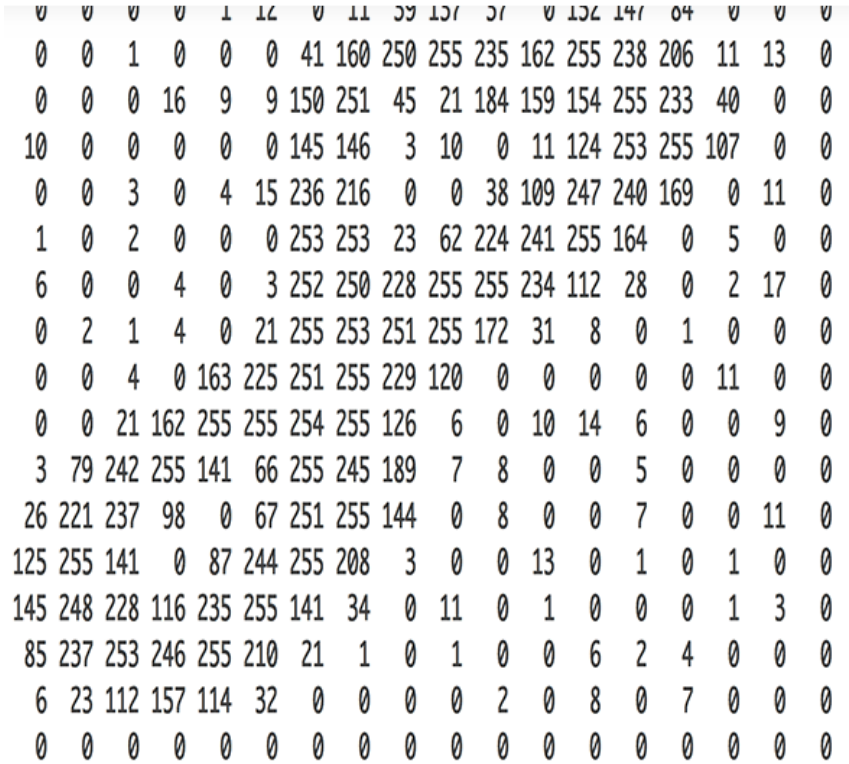
An image is essentially a matrix (two- or more dimensional) of real numbers.

- Greyscale representation: Single channel representation



The numeric matrix for this image is:

Images II



0	0	0	0	1	12	0	11	39	157	37	0	152	147	84	0	0	0
0	0	1	0	0	0	41	160	250	255	235	162	255	238	206	11	13	0
0	0	0	16	9	9	150	251	45	21	184	159	154	255	233	40	0	0
10	0	0	0	0	0	145	146	3	10	0	11	124	253	255	107	0	0
0	0	3	0	4	15	236	216	0	0	38	109	247	240	169	0	11	0
1	0	2	0	0	0	253	253	23	62	224	241	255	164	0	5	0	0
6	0	0	4	0	3	252	250	228	255	255	234	112	28	0	2	17	0
0	2	1	4	0	21	255	253	251	255	172	31	8	0	1	0	0	0
0	0	4	0	163	225	251	255	229	120	0	0	0	0	0	11	0	0
0	0	21	162	255	255	254	255	126	6	0	10	14	6	0	0	9	0
3	79	242	255	141	66	255	245	189	7	8	0	0	5	0	0	0	0
26	221	237	98	0	67	251	255	144	0	8	0	0	7	0	0	11	0
125	255	141	0	87	244	255	208	3	0	0	13	0	1	0	1	0	0
145	248	228	116	235	255	141	34	0	11	0	1	0	0	0	1	3	0
85	237	253	246	255	210	21	1	0	1	0	0	6	2	4	0	0	0
6	23	112	157	114	32	0	0	0	0	2	0	8	0	7	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- RGB (Red-Green-Blue) representation: Three channel representation.
  - This means each image can be written as a three dimensional matrix.
  - The depth is 3 (each depth represents a channel)

Images III

However, how do we convert this numeric matrix to a feature vector?

- Converting the whole matrix to a vector (row-major)
  - Useful for standard ML tools such as: Decision Tree, Random Forest, NB, SVM etc.
  - Merit: Easy to transform to 1D
  - Demerit: Loss of spatial information of pixels and interpixel relationships
- Feature extraction using Convolution:
  - Inherent in Convolutional Neural Networks
  - Merit: Preserves spatial relationships
  - Demerit: Computation heavy

(Attend ML or DL Labs to know more!)

Audio I

- Sound can be represented in many different machine readable format.
  - wav (Waveform Audio File) format
  - mp3 (MPEG-1 Audio Layer 3) format
  - WMA (Windows Media Audio) format
- Short-term feature extraction
  - The input signal is split into short-term windows (frames) and computes a number of features for each frame.
  - This feature leads to a sequence of short-term feature vectors for the whole signal.
- Mid-term feature extraction
  - The signal is represented by statistics on the extracted short-term feature sequences described above.

Audio II

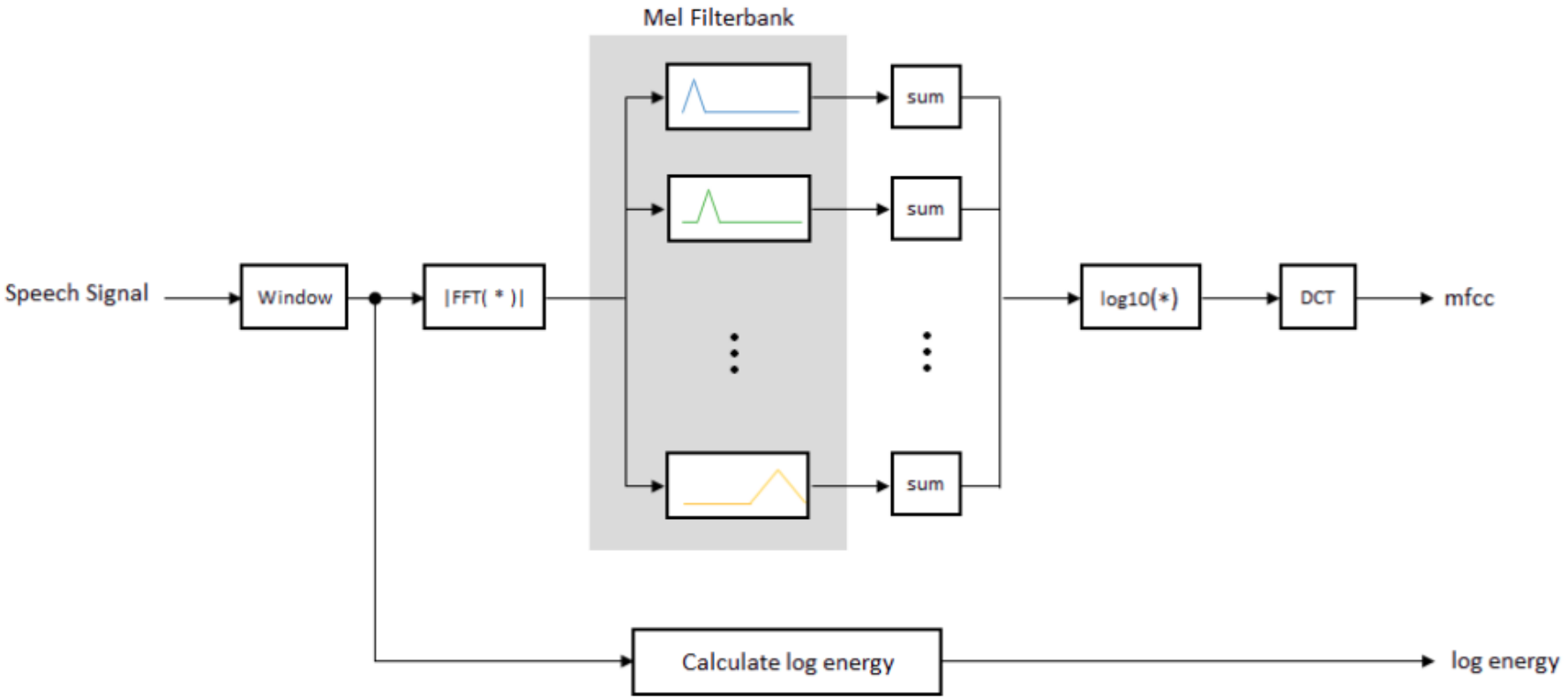
Here, we will look at one well-known audio feature extraction method called **Mel Frequency Cepstral Coefficents (MFCCs)**.

Steps:

- Frame the signal into short frames
- For each frame calculate the periodogram estimate of the power spectrum
- Apply the mel filterbank to the power spectra, sum the energy in each filter
- Take the logarithm of all filterbank energies
- Take the DCT of the log filterbank energies
- Keep DCT coefficients 2-13, discard the rest.

The motivating idea of MFCC is to *compress information* about the *vocal tract (smoothed spectrum)* into a small number of *coefficients* based on an understanding of the cochlea (the spiral cavity of the inner ear).

Audio III



(Source: https://in.mathworks.com/help/audio/examples/speaker-identification-using-pitch-and-mfcc.html)

Link Prediction

- ML Problem: Studying networks to predict the emerging interactions
- The problem of predicting future or missing relationships in networks is called *link prediction*.
- Solution: Treat it as a *clustering* or *classification* task
- Major issues:
  - The graph can be too sparse (e.g. a network of researchers in variety of fields)
  - The graph can be too dense (e.g. a social network)

Graphs II

Feature extraction methods for *link prediction* problem:

- Similarity based methods: If two nodes are more similar, they are more likely to be linked in the future.
  - Global approaches: Use the whole topology of the network to rank similarity between node pairs
    - Katz Index
    - Leicht-Holme-Newman
    - SimRank
    - Pseudo-inverse Laplacian
    - Rooted PageRank
    - Escape Probability
    - Random Walk
    - Blondel Index
    - Matrix Forest Index
    - Maximal Entropy RW
    - Random Walk with Restart

Graphs III

- Local approaches: If node pairs have common neighbors structures or one of them already has a significantly higher degree, they will probably form a link in the future
  - Common Neighbors
  - Jaccard Index
  - Salton Index
  - Sorensen Index
  - Preferential Attachment
  - Adamic-Adar Index
  - Resource Allocation
  - Functional Similarity Weight
  - Local Neighbors Link Index
  - CAR Based Index
  - Mutual Information Index
  - Individual Attraction index
  - Parameter Dependent Index
  - Leicht-holme-Newman
  - Hub Depressed Index
  - Hub Promoted Index

Graphs IV

- Quasi-local approaches: A set of approches emerging from the usage of whole graph strcture (from global approaches) and low time-complexity methods (from local approaches)
  - Local Path Index
  - Local RW
  - Superposed RW
  - FriendLink
  - PropFlow Predictor

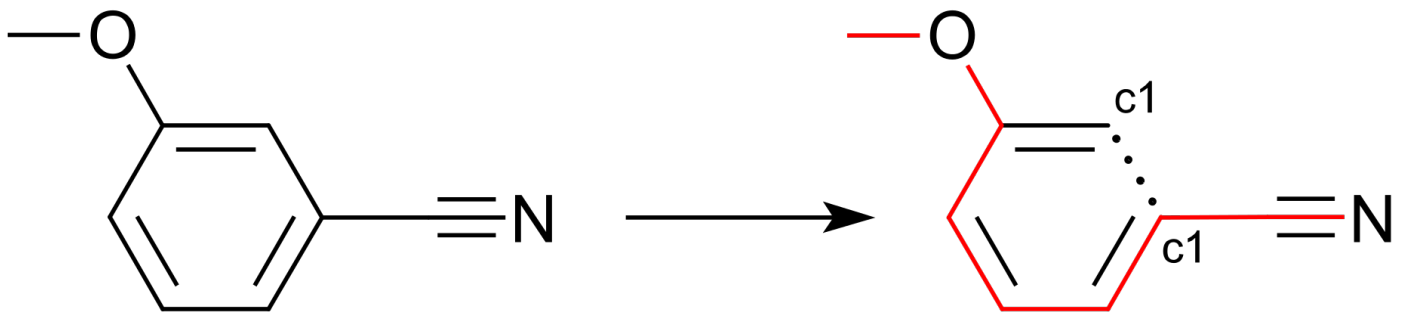
Graphs V

Prediction problems drug design:

- ML problem: Analyse molecular structure to predict: Quantitative Structure-Activity Relationships (QSAR)
- Example: Predicting the toxicity level of a molecule
- Solution: Treating the problem as a *Regression problem*. Activity $= f$(physiochemical properties and/or structural properties) $+$ *error*

Graphs VI

- Major issues:
  - Input may only be a SMILE string (a high level representation molecule: a string obtained by printing the symbol nodes encountered in a depth-first tree traversal of a chemical graph.)



3-cyanoanisole : COc(c1)cccc1C#N.

  - All physiochemical properties (may not be known
  - All structural properties (e.g. rings, connected rings, fused rings, and various functional groups) are not known

Graphs VII

Structural features extraction method:

- Use vast amount of knowledge already known in the field (called, Domain Knowledge)
  - What are various functional groups?
  - When to say the molecule has a functional group?
  - What is a ring?
  - What are fused rings?
  - What are connected rings?
- Produce the above with the information already known from the inputs (atoms and bonds)

Popular ML methods for this: Inductive Logic Programming[2]

[2] Srinivasan, A., & King, R. D. (1999). Feature construction with inductive logic programming: A study of quantitative predictions of biological activity aided by structural attributes. Data Mining and Knowledge Discovery, 3(1), 37-57.

Questions

What would be your approach, when:

- Time-series (include temporal dependence)
- Inputs are videos (includes spatial and temporal dependence)