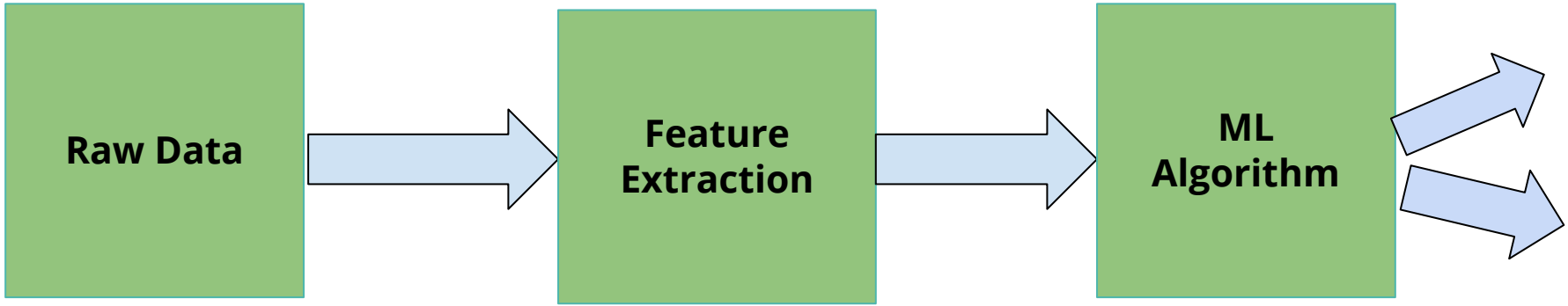
Master Lecture II

Pipeline



Nearest Neighbour

Decision Tree

Linear Classifier

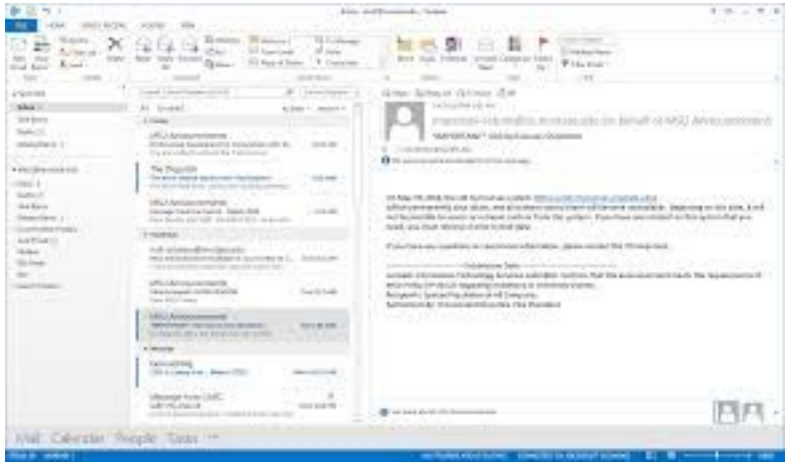
Feature Engineering

Representation

Three Generations of Representations/Features

- Intuition driven representation
- Representations derived based on
  - Statistics, Signal Processing etc.
- Representations that can be learned

Problem: Classify Textual Content



Emails

Web Page

Problem: Classify into one of "C" classes.
Eg. "Spam vs non-spam", "Professional vs Personal",
"Movies Vs Sports Vs Politics"

What do we mean by representing text?

- Representations for “words”?
- Representations for “documents”?

Bag of Words - Text Domain

- Orderless documentation representation, frequencies of words from a dictionary.

Bag of Words - Text Domain

- Orderless documentation representation, frequencies of words from a dictionary.

Bag of Words - Text Domain

- Orderless documentation representation, frequencies of words from a dictionary.

1-Hot Representation and Histograms

D1 : Andrew is intelligent.
D2 : Ram is a good boy. Ratna is also good.

A	1 0 0 0 0 0 0 0 0
Also	0 1 0 0 0 0 0 0 0
Andrew	0 0 1 0 0 0 0 0 0
Boy	0 0 0 1 0 0 0 0 0
Good	0 0 0 0 1 0 0 0 0
Intelligent	0 0 0 0 0 1 0 0 0
Is	0 0 0 0 0 0 1 0 0
Ram	0 0 0 0 0 0 0 1 0
Ratna	0 0 0 0 0 0 0 0 1

1-Hot representations

1-Hot Representation and Histograms

Histogram is the sum of all one hot representations of all the words

D1

0 0 1 0 0 0 0 0 0 + 0 0 0 0 0 0 1 0 0 + 0 0 0 0 0 1 0 0 0

Andrew **Is** **Intelligent**

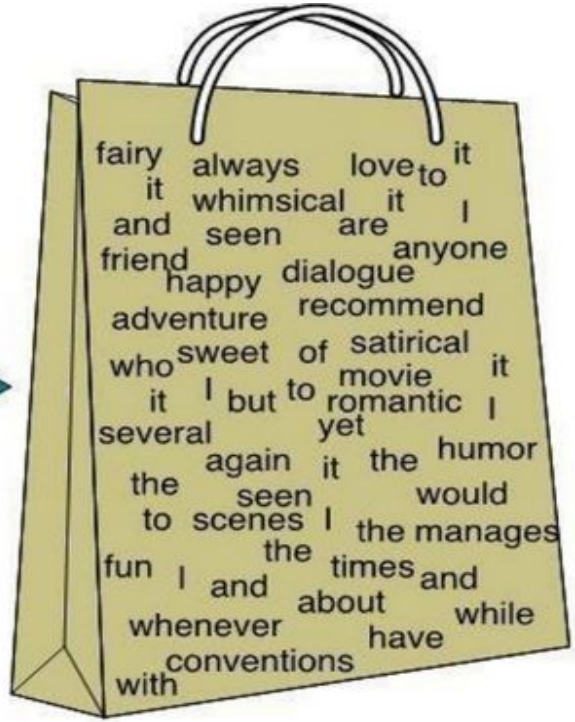
0 0 1 0 0 1 1 0 0

Histogram for both the documents :

	A	Also	Andrew	Boy	Good	Intelligent	Is	Ram	Ratna
D1	1	1	0	1	2	0	2	1	1
D2	0	0	1	0	0	1	1	0	0

Histogram of Word Occurrences

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

Comments: Weight Words (eg TF-IDF)

- Not all words are equally useful.
- Stop words: Words that will not add value. (can be removed)
  - Eg. “The” , “of”, “and”
- Frequent words (TF: Term Frequency)
  - Higher the frequency, more useful
- Rare across documents (IDF: Inverse Document Frequency)

Comments: Weight Words (eg TF-IDF)

- Weight words: Proportional to TF and Inv. Prop to IDF.
- (In the lab, you just remove some frequent and some rare words.)

Word2Vec

Consider two sentences

- Gandhi went on a march at Delhi
- Gandhi went on a strike at Delhi.

- Their representation is different since machine does not understand “march” has something to do with “strike”.

Consider two sentences

- Gandhi went on a march at Delhi
- Gandhi went on a strike at Delhi.

- Let us represent these as triplets (after removing stop words):
- <Gandhi, March, Delhi> and <Gandhi, Strike, Delhi>
- In general <a1,a2,a3>
- (We will relax the three word sentences later.)

What do we want?

- We want the representation of Strike and March to be the same (or at least very similar).

- How?

What do we want?

- We want the representation of Strike and March to be the same. How?
- Ans: Let us learn such a “nice” “magic” representation. (details follow)
- Intuition: We know that meaning of a word is defined by the co-occurring words.

Example (detour)

Marco saw a furry little wampimuk hiding in the tree.

What is Marco? A person.

What is wampimuk?

What do we want? Ans:

- Learn a representation such that:
- **Representation of the middle word is sum of representations of the left and right words.**

$$\left[R(a2)\right] = \left[R(a1)\right] + \left[R(a3)\right]$$

For all sentences (assume sentences are triplets)

What do we want? Ans:

- Learn a representation such that:
- **Representation of the middle word is function of representations of the left and right words.**

$$[R(a2)] = f([R(a1)], [R(a3)])$$

For all sentences (assume sentences are triplets)

What do we want? Ans:

- Learn a representation such that:
- **Representation of the middle word is function of representations of the left and right words.**

$$[R(a2)] = f(\mathbf{W}, [R(a1)], [R(a3)])$$

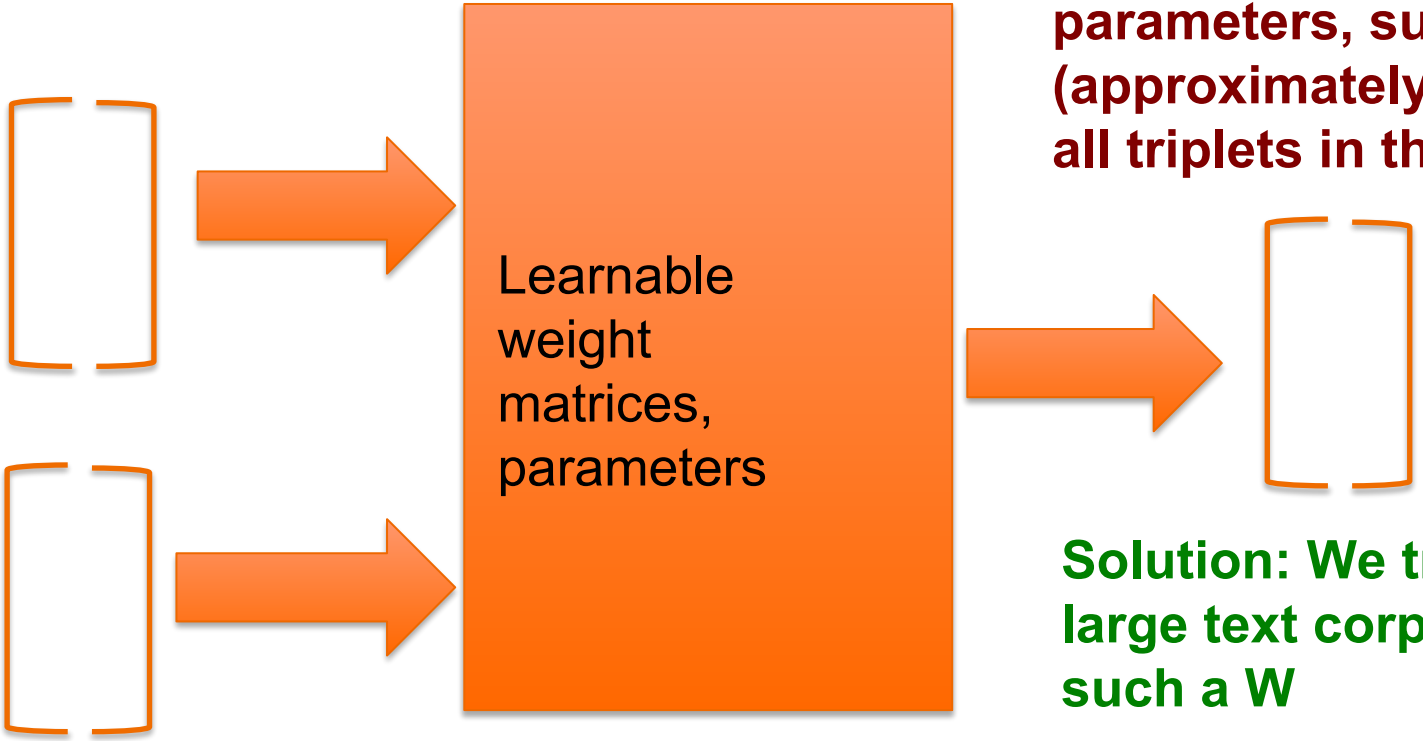
For all sentences (assume sentences are triplets)

(You also need a learnable parameters/weights, W (matrices))

Word2Vec: Comments

- Meaning of word defines the co-occurrence of words
- Example
  - I travelled by Bus. I travelled by Car
- Bus and Car should have similar representations but not identical. Why?
- Example
  - This is our family Car. This is a public Bus.

Pictorially



Problem: Learn the parameters, such that this is (approximately) "TRUE" for all triplets in the text.

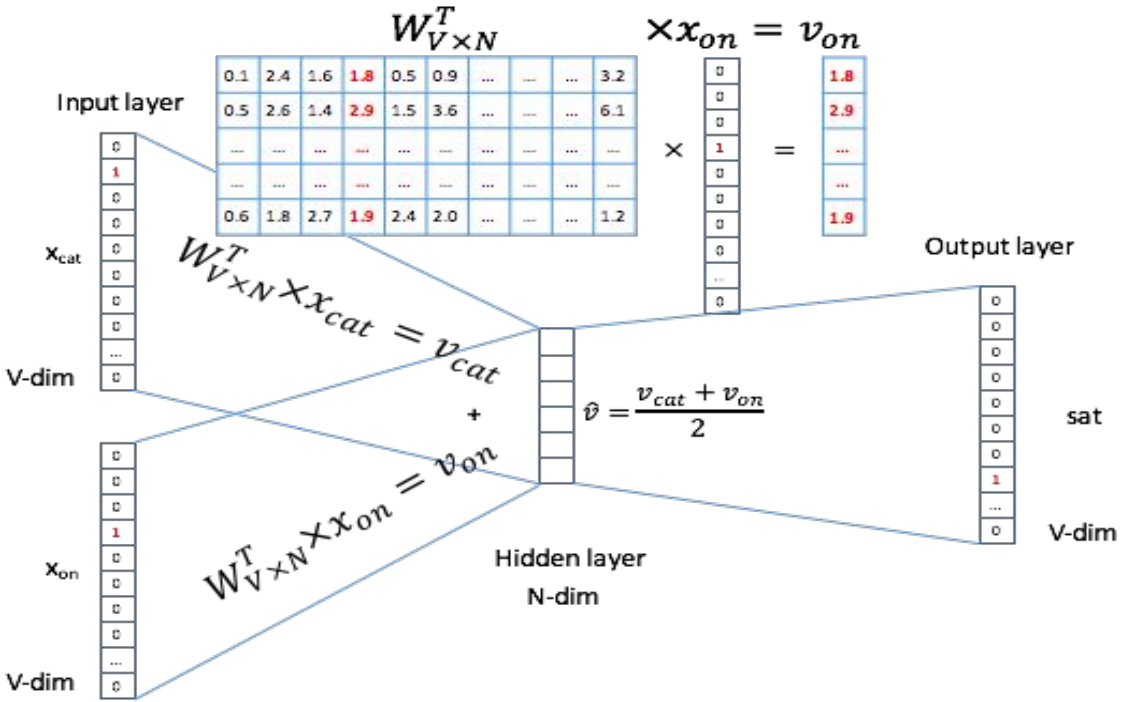
Solution: We train with a large text corpus and find such a W

Detail

- Weight matrices are of size
- **W** is V X d

- V is the vocabulary size; d is the size of the new representation
- For the ith word in the vocabulary, representation is the ith row of the matrix.

Detail (eg. <cat, sat, on>)



1-Hot to Rich Representations

book [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]

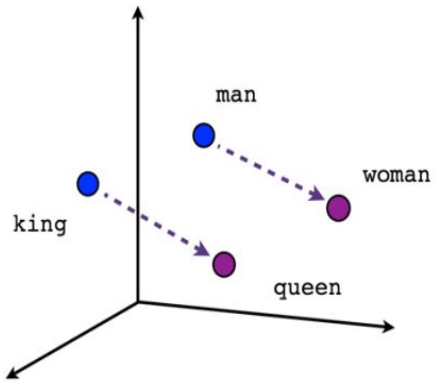
library [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0]

Word	w	$C(w)$
"the"	1	[0.6762, -0.9607, 0.3626, -0.2410, 0.6636]
"a"	2	[0.6859, -0.9266, 0.3777, -0.2140, 0.6711]
"have"	3	[0.1656, -0.1530, 0.0310, -0.3321, -0.1342]
"be"	4	[0.1760, -0.1340, 0.0702, -0.2981, -0.1111]
"cat"	5	[0.5896, 0.9137, 0.0452, 0.7603, -0.6541]
"dog"	6	[0.5965, 0.9143, 0.0899, 0.7702, -0.6392]
"car"	7	[-0.0069, 0.7995, 0.6433, 0.2898, 0.6359]

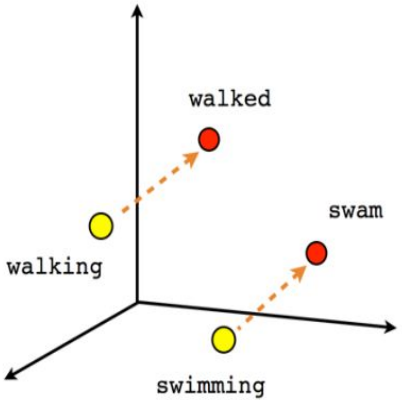
$$\text{king} - \text{man} + \text{woman} = \text{queen}$$

Mikolov 2013

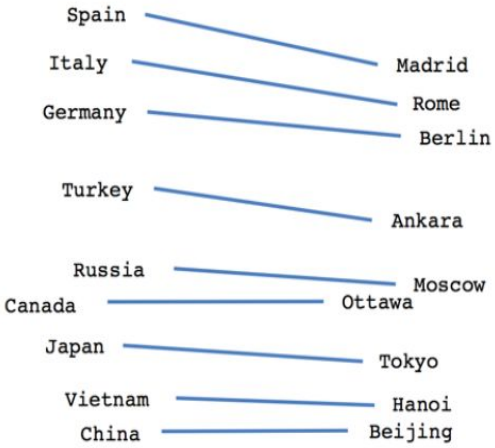
Examples



Male-Female



Verb tense

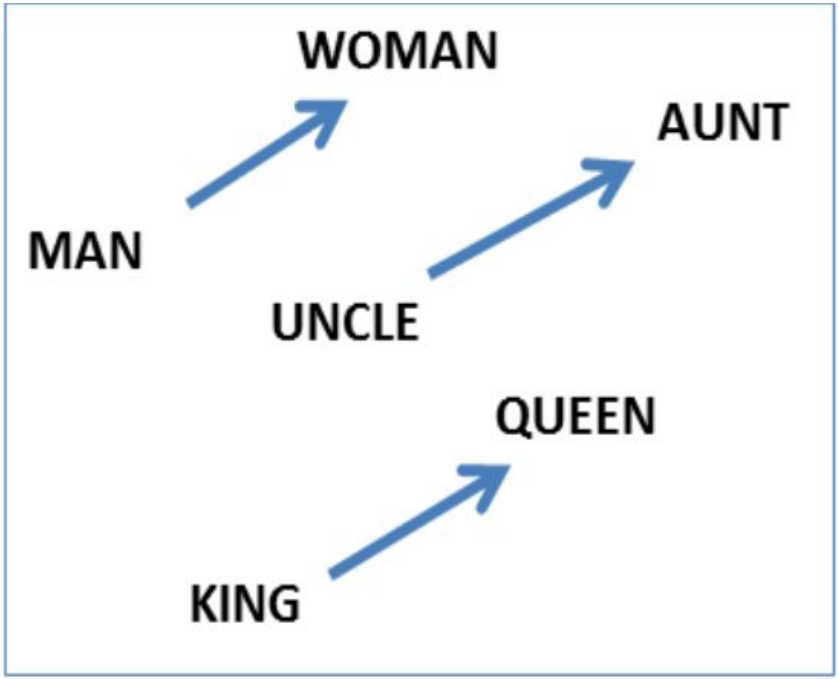


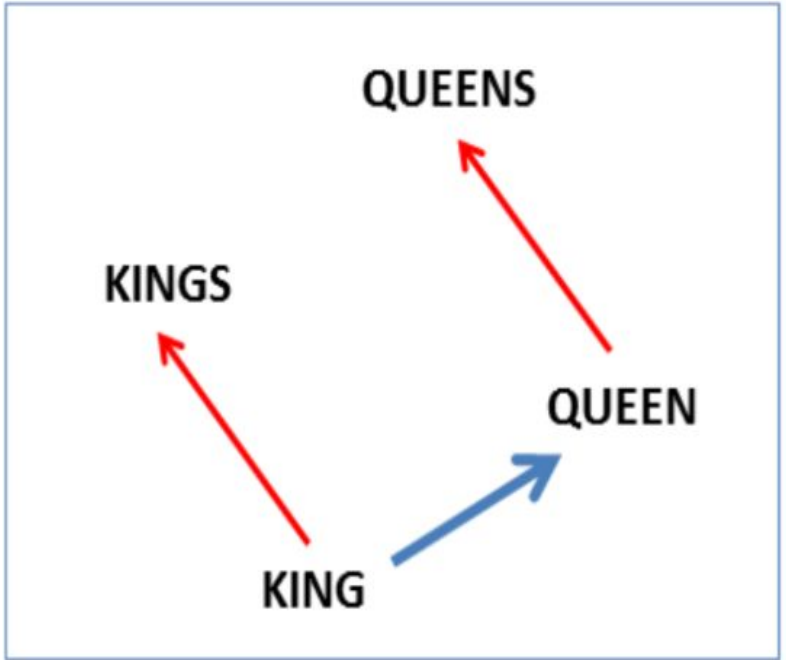
Country-Capital

Examples

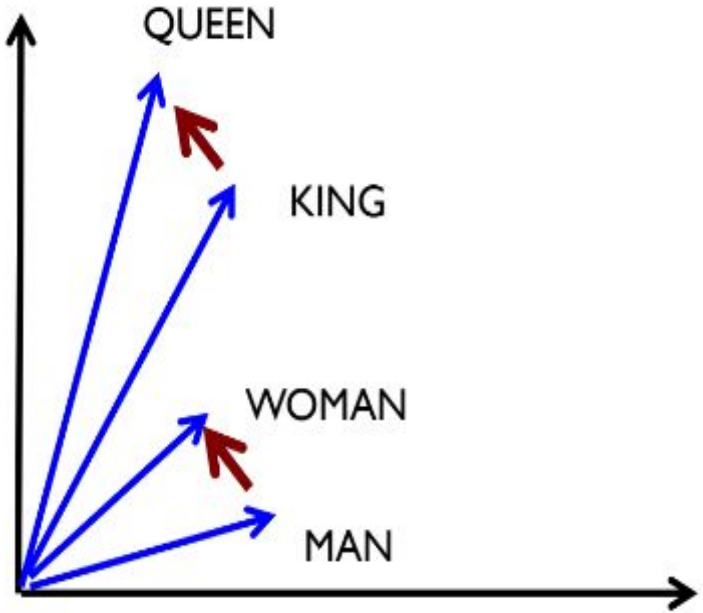
- I walked 10 kms
- I was walking 10kms
- I swam 10kms
- I was swimming 10kms

Word Similarity = Vector similarity





Visualising "Word-Arithmetic"



- (This fancy arithmetic is known earlier also.It became popular with word2vec)

$$vector[Queen] = vector[King] - vector[Man] + vector[Woman]$$

Representing documents (email/web page)

- Representation for the document can be now the sum of word2vecs

More Examples

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

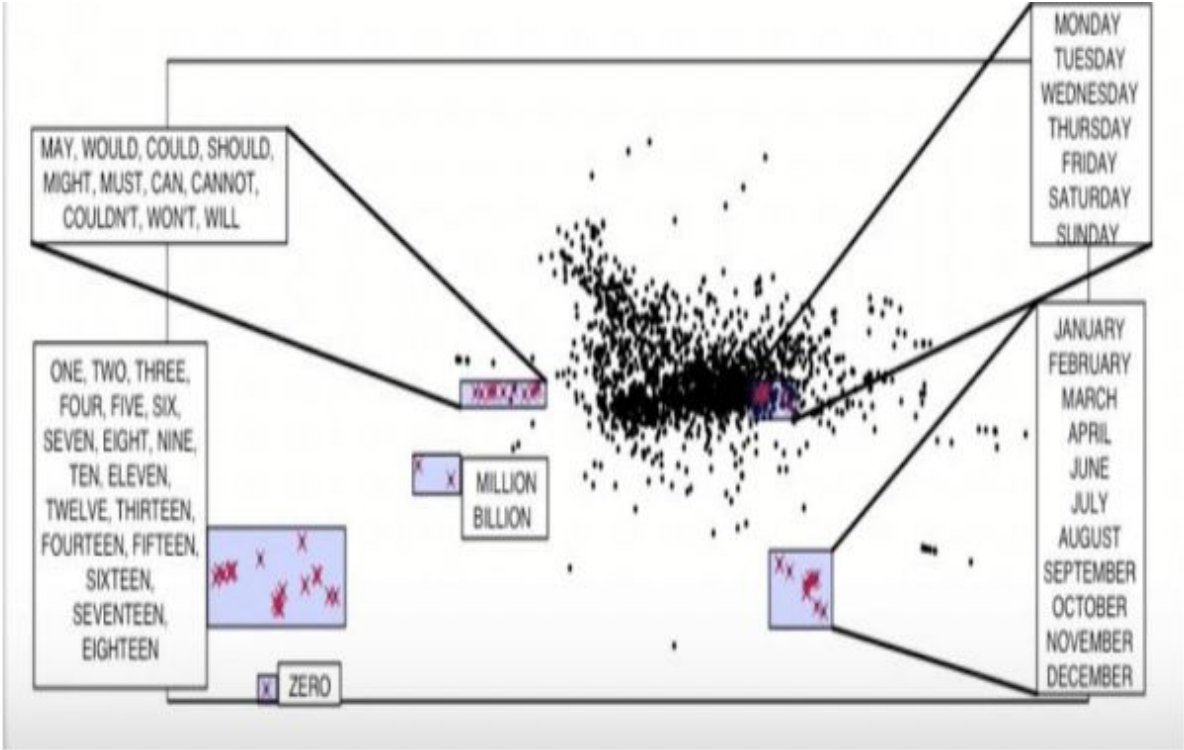
What words have embeddings closest to a given word? From Collobert *et al.* (2011) (http://arxiv.org/pdf/1103.0398v1.pdf)

More Examples

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Relationship pairs in a word embedding. From Mikolov *et al.* (2013b) (http://arxiv.org/pdf/1301.3781.pdf).

More Examples

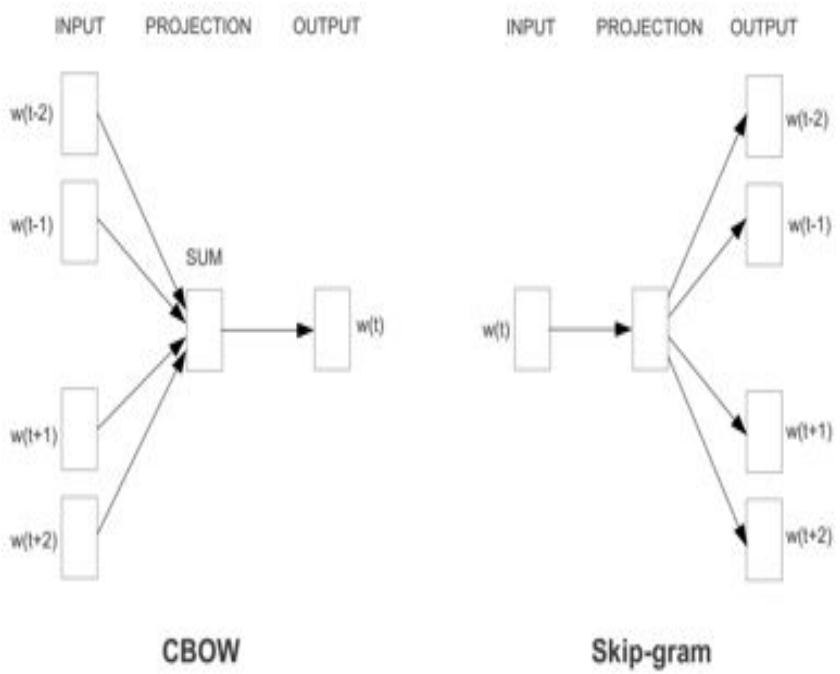


From: Blitzer et al. 2014

Variants

- Continuous Bag of Word (CBOW): use a window of word to predict the missing word

- Skip-gram (SG): use a word to predict the surrounding ones in the specified window.



Demo

Comment: Categorical Variables

- We are worried that we also have "Categorical Variables"
  - Text is classical example
- A 1-hot representation is a possibility
- However, we can now have Dense embedding/representations
  - Motivated by Word2Vec

Some Questions

- How do I use/ extend in my domain?
  - Every problem has its own characteristics. Let us not assume that we can copy solutions.
  - There are numerous examples of generic notions getting employed or extended to your notion. (please see next slide)

Representing Complex Entities

- Products :
  - Product has attributes
- **Represent items as vectors in the same space and use their distances to compute recommendations**
- Many recent works :
  - Prod2Vec : Sequence of Products
  - MetaProd2Vec : (categories, brand, tags etc ..)

Representing Complex Entities

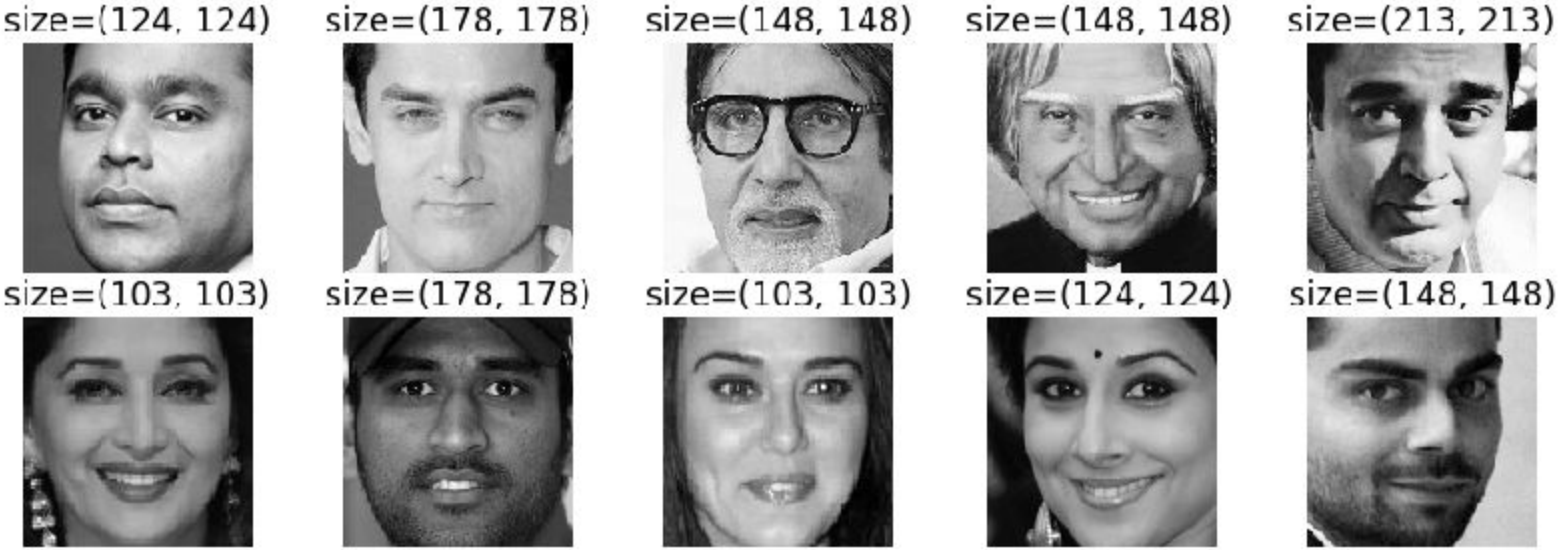
- Medical Records
  - Visits, Numerical attributes, Personal attributes, Images, Reports
- Data:
  - Visits: The sequence in which a patient goes to a hospital
  - Medical terms (diagnosis, disease, medication, procedure)

“X”2Vec

- Motivated by word2vec
  - Node2vec
  - Doc2vec
  - See:https://github.com/MaxwellRebo/awesome-2vec
- Learn from domain examples
- An embedding that can be manipulated

Image Representation

Recognize Indian Celebrities



Step 0: Preprocess to 224 X 224 or 50716 X 1

Representations

- Eigen faces(classic)
  - (Same as PCA you already know)

- VGG Deep Net Features(Modern)
  - (the same you had seen for CFAR)

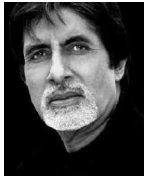
Visualizing Mean as an Image

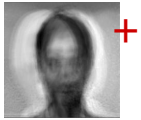
Mean of faces $\frac{\sum_i X_i}{N}$

Visualizing Eigen Vectors as Images

Top 20 Eigen Vectors (eigen Faces)

Representing with EigenFaces



= 6.8 *  + 3.5 *  + 2.89 *  -1.2 *  .....+0.0002 * 

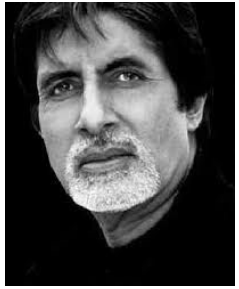
Any face in the database can be represented as a linear combination of the eigen faces.

Any face can be done so !!

 = 5.15 *  1.5 *  0.59 *  - 0.2 *  .....- 0.0006 *

Image can be represented as

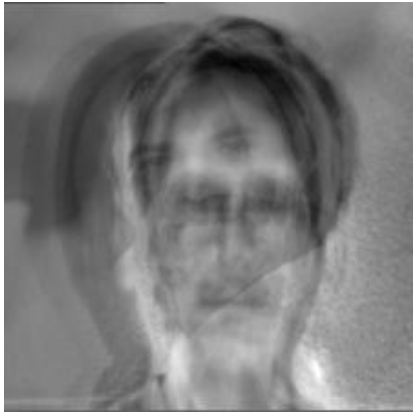
$$\begin{bmatrix} 5.15 \\ 1.5 \\ 0.59 \\ -0.2 \\ . \\ . \\ . \\ . \\ 0.0006 \end{bmatrix}$$



$$\begin{bmatrix} 6.8, \\ 3.5, \\ 2.89, \\ 1.2, \\ . \\ . \\ . \\ . \\ 0.00002 \end{bmatrix}$$

Reconstruction

- Using only a few eigen faces out of the n eigen faces



Using 10 eigenfaces

Using 30 eigenfaces

Using 50 eigenfaces

Using 165(all) eigenfaces

165 eigenfaces in total in this case

Reconstruction

Reconstructed Image = Mean Image + Dot product of Weights and Eigen faces

Which is the reconstructed image?

Reconstructed Image

Original Image

PCA/Eigen Face Algorithm: Detail

1. Compute the mean feature vector

   $\mu = \frac{1}{p}\sum_{k=1}^{p} x_k$, where, x_k is a pattern ($k = 1$ to p), p = number of patterns, x is the feature matrix

2. Find the covariance matrix

   $C = \frac{1}{p}\sum_{k=1}^{p}\{x_k - \mu\}\{x_k - \mu\}^T$ where, T represents matrix transposition

3. Compute Eigen values λ_i and Eigen vectors v_i of covariance matrix

   $Cv_i = \lambda_i v_i$ $\quad (i = 1, 2, 3, \ldots q)$, q = number of features

4. Estimating high-valued Eigen vectors

   (i) Arrange all the Eigen values (λ_i) in descending order

   (ii) Choose a threshold value, θ

   (iii) Number of high-valued λ_i can be chosen so as to satisfy the relationship

   $\left(\sum_{i=1}^{s}\lambda_i\right)\left(\sum_{i=1}^{q}\lambda_i\right)^{-1} \geq \theta$, where, s = number of high valued λ_i chosen

   (iv) Select Eigen vectors corresponding to selected high valued λ_i

5. Extract low dimensional feature vectors (principal components) from raw feature matrix.

   $P = V^T x$, where, V is the matrix of principal components and x is the feature matrix

In Practice

- The images need not be aligned very nicely
- Mean and Eigen vectors need not be this “beautiful”

Demo

Deep VGG Features

Deep Learning = End to End Learning (Raw data to labels)
Deep Learning = Feature Learning!!

Raw Image

Deep Neural Networks
Many linear and nonlinear operations

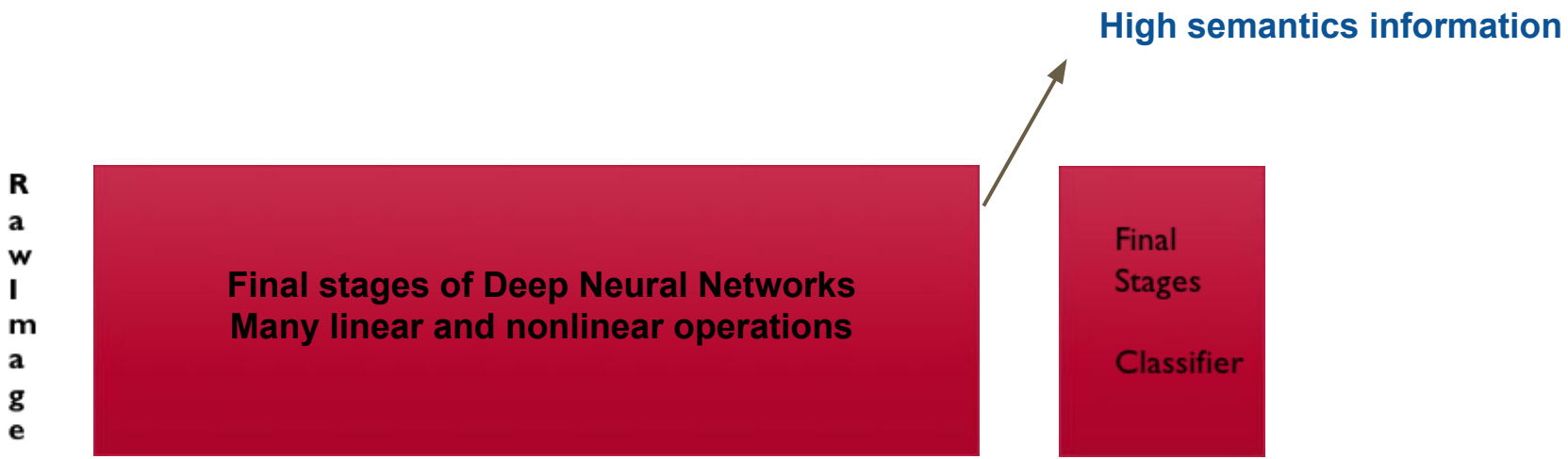
Final Stages

Classifier

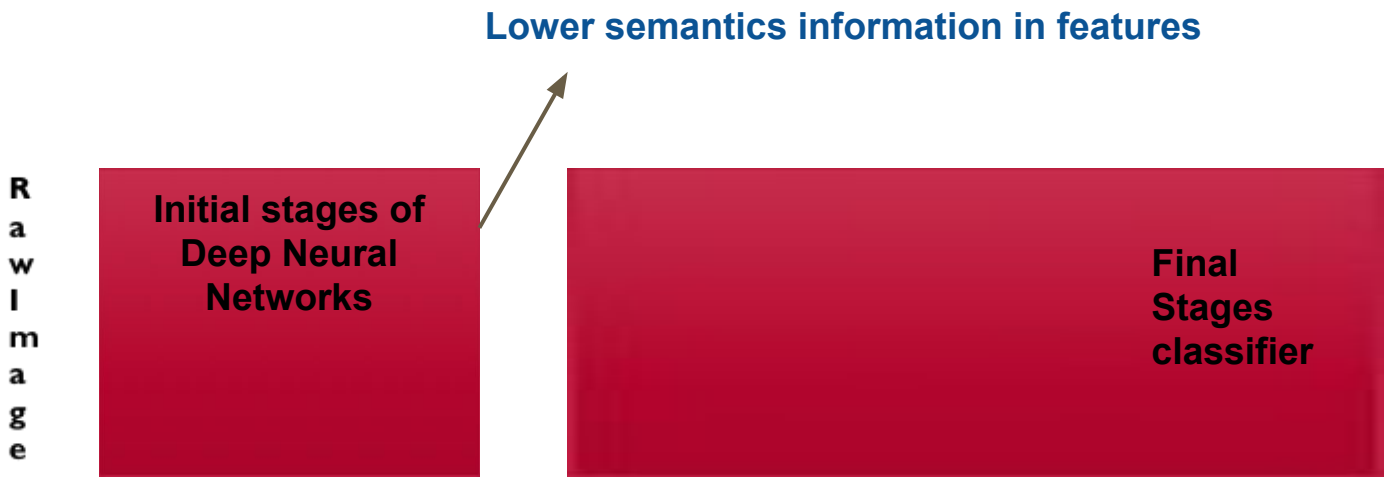
1000 Labels For a 1000 class classification

An intermediate representation from a popular "Deep VGGNet", which was designed and trained for solving a "general" 1000 class classification.

Deep VGG Features



Deep VGG Features



Speech

(Brief Explanation)

Example Problem

- Sound waves (.wav files)
- 30 short commands (“cat”, “dog”, “go”, “happy”)
- 1 sec duration
- 65000 samples (many people)

Cat

Dog

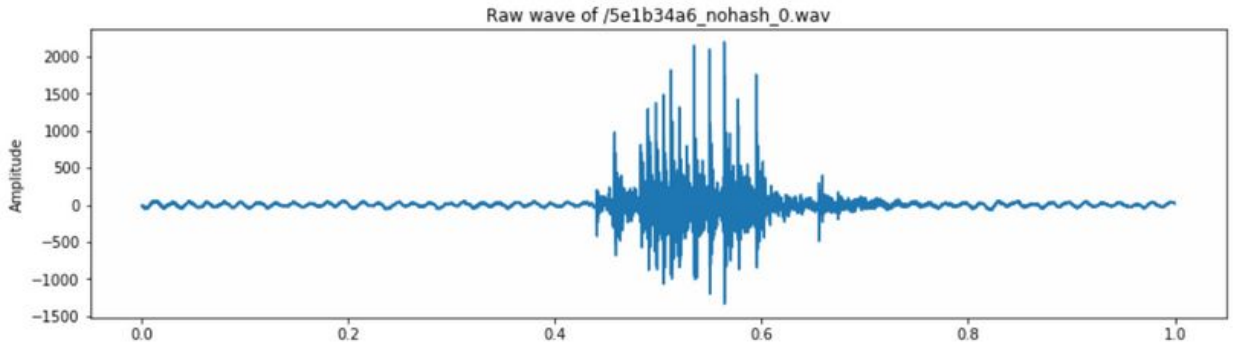
Go

Happy

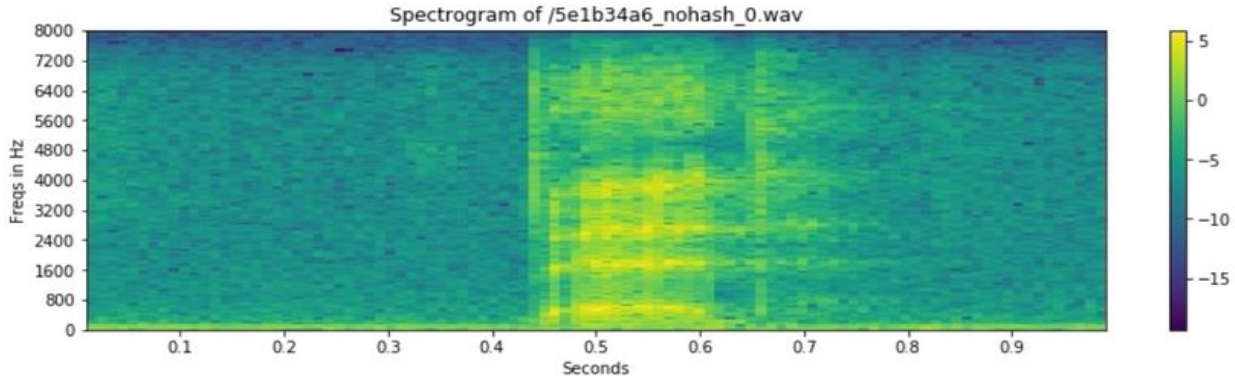
Representations

- A : MFCC (Signal processing based; Classical)
  - Mel Frequency Cepstral Coefficients
- B : CNN Based (Modern)
  - VGG Features on the Mel Spectrogram

Classical Feature (MFCC)

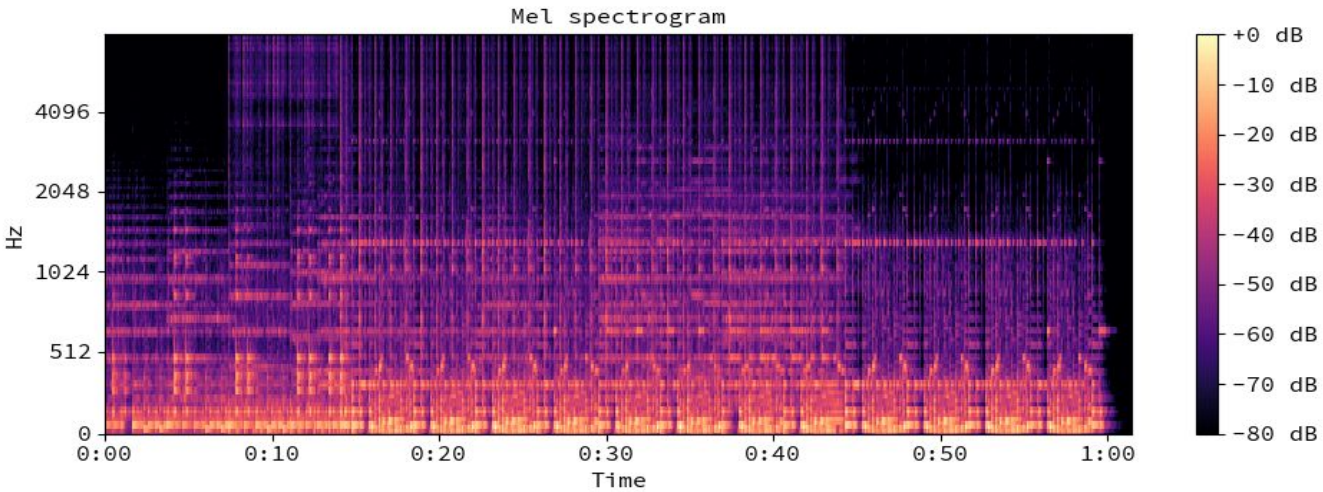


Amplitude Vs Time



Frequency Vs Time

Features from Mel Spectrogram



MFCC
(Hand coded Classic Features)

VGG19-Features
(Trained on Mel spectrograms)

Linear Models

The Machine Learning Framework

$$Y = f(x)$$

Output — **prediction function** — **feature representation**

Training: given a training set, estimate the prediction function **f** by minimizing the prediction error

Testing: apply **f** to unknown test sample ***x*** and predicted value(output) is **y**

Parameter: Primary problem is to find the parameters w

Sample and Representation

A sample is easy to visualize in 2D

$$x = (x, y) or (x_1, y_2) \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

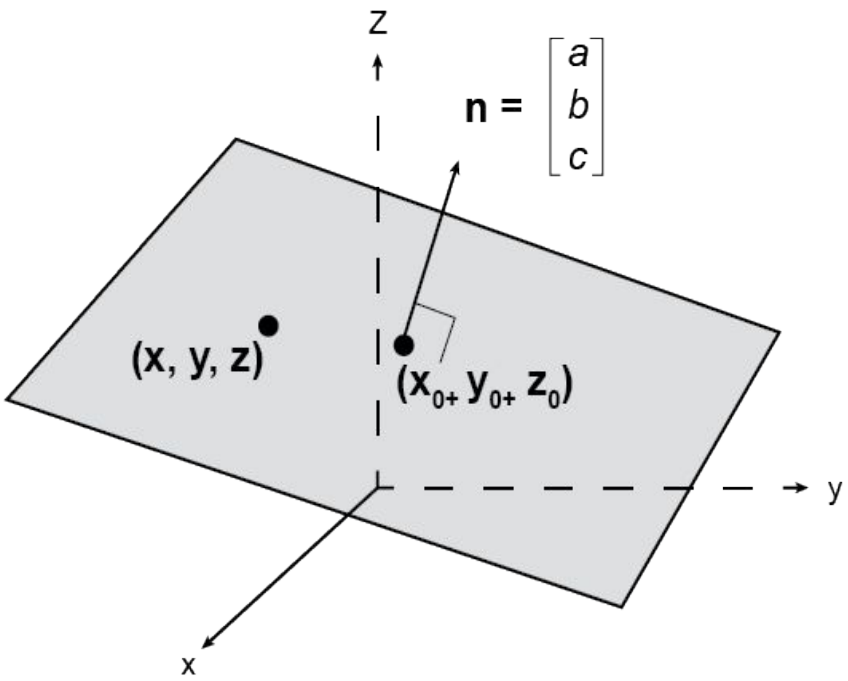
and sometime in 3D with some effort $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

And we often need much larger dimensionality in practice

$$x = \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_{100} \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_d \end{bmatrix}$$

Linear Model

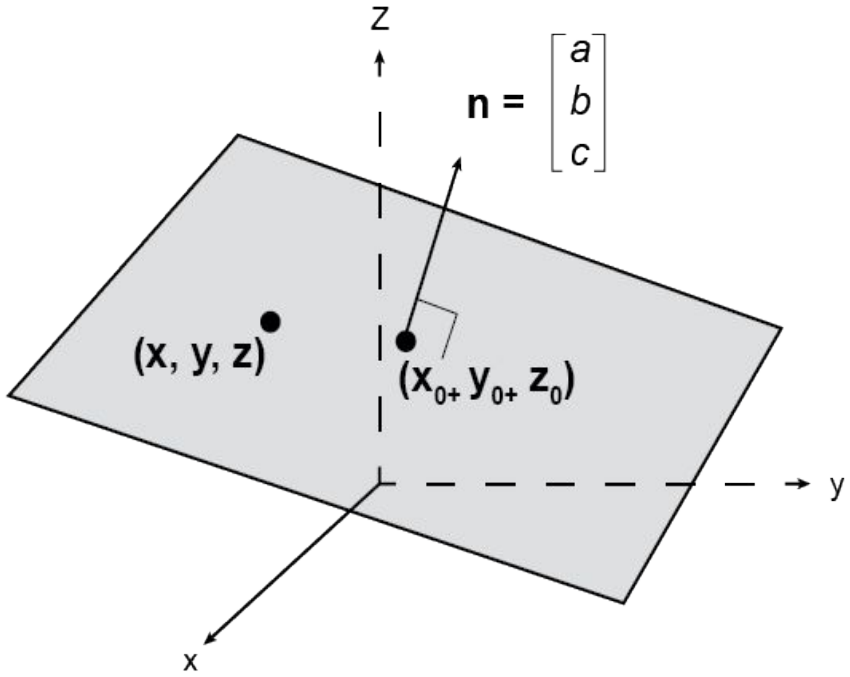
- Linear when f is a line
- In general,a hyperplane
  - $y = ax + b$
    - a, b : parameters of the model
  - $y = w^T x = w.x$ when multivariate
    - $x = [1\ x_1\ x_2\ \ldots\ x_d]^T$
    - w : parameters of model



Linear Model

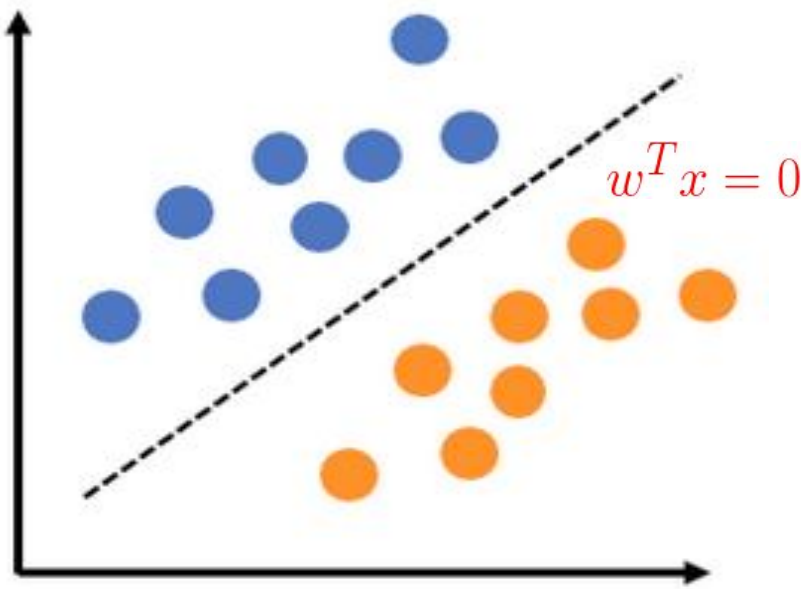
- Line: Only 2 parameters in 2D. d parameters in a d-dimensional space

$$y = f_w(x) = w^T x$$



Linear Classifier

- It has linear partition or decision boundaries.



Decision Boundary

- Decision boundary : Hyperplane

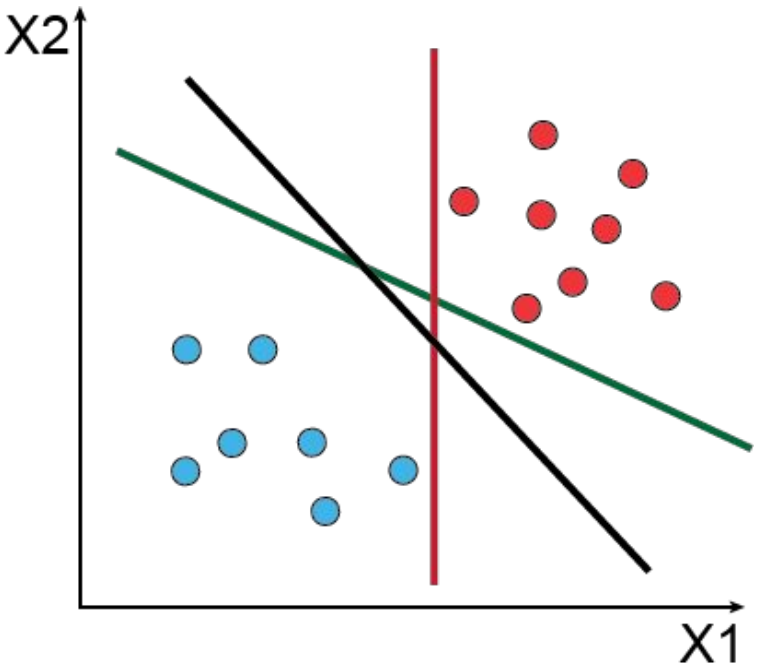
$$w^T x = 0$$

- Class 1 lies on the positive side $w^T x > 0$
- Class 0 lies on the negative side. $w^T x < 0$

- (This is discontinuous.)

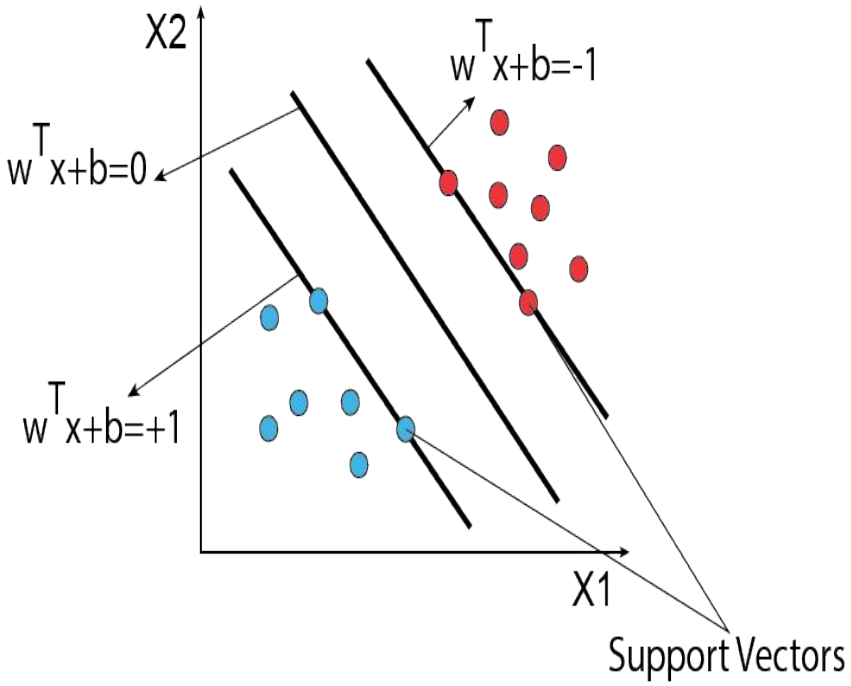
Boundary with Margin

- Several decision boundaries are possible



Support Vector Machines

- SVMs maximize the *margin* around the separating hyperplane.
  - A.k.a. large margin classifiers
- The decision boundary is fully specified by a subset of training samples, *the support vectors*.



The Problem

Find w, given examples $(x_i, y_i), i = 1, 2, ..., n$

Supervised situation: output label is available for all n training samples

- Objective: predict y values for input values x that are not seen before
  - Called generalization in Machine Learning
- How do we find w? Gradient descent!

Objective function

- **Error** or **loss**: How far is the prediction from the observed value?

$$Total\,Loss\;\; J = \sum_{AllSamples(x,y)} L(f(x,w),y)$$

- **Strategy**: Bring the predicted value closer to the observed value by adjusting w

Example: Loss

- For example: Loss could be the squared difference of predicted and actual values

$$L = (y - f(x, w))^2$$

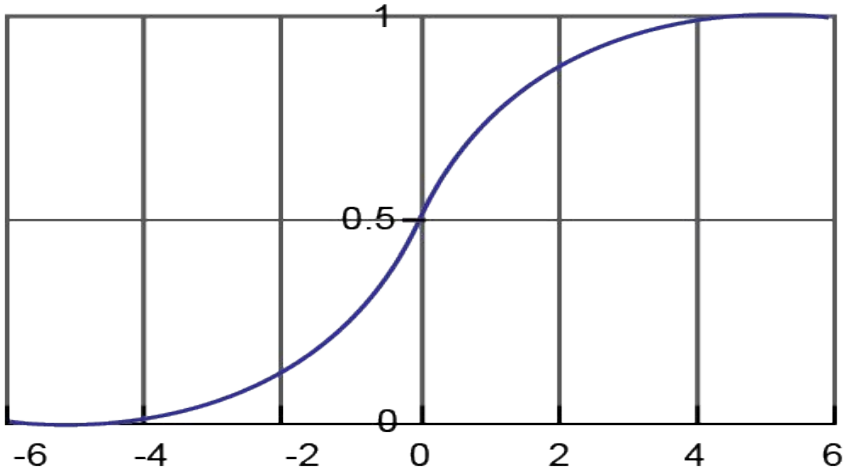
- Final objective is to minimize J, which is sum of L for all samples.

Logistic Regression

- To make the loss "differentiable" (see next few slides)

$$\frac{1}{1+e^{-w^T x}}$$

- (Nice probabilistic interpretation)



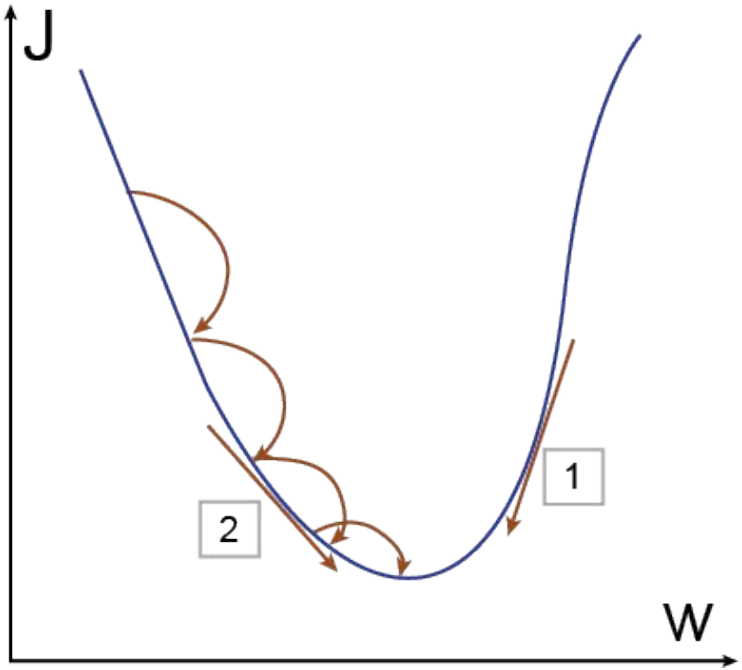
Gradient Descent

Optimization Algorithm

Gradient Descent

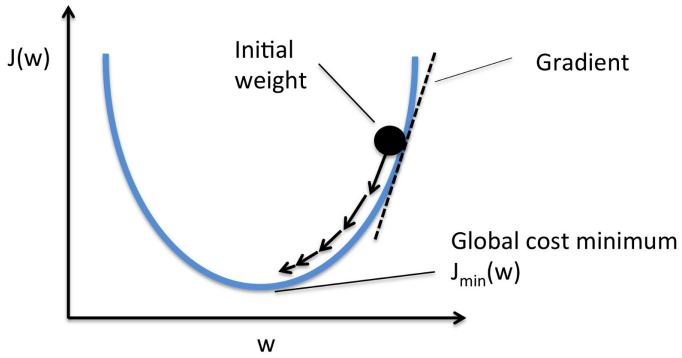
- Minimum of the function lies in the opposite direction of gradient
- Start with a guess
- Take a step against gradient:

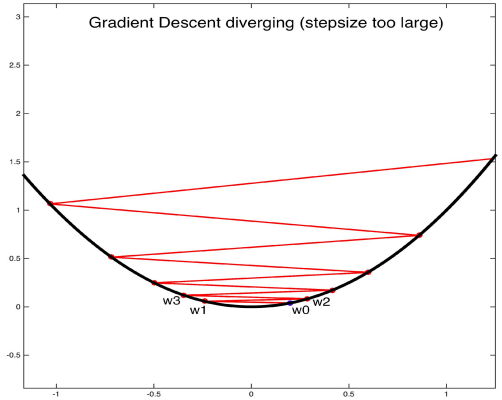
$$w' = w - \eta \Delta J(w)$$



Gradient Descent - Learning rate

- Learning rate is critical
  - Start high to make rapid strides
  - Reduce for smoother converge





Gradient Descent - Initialization

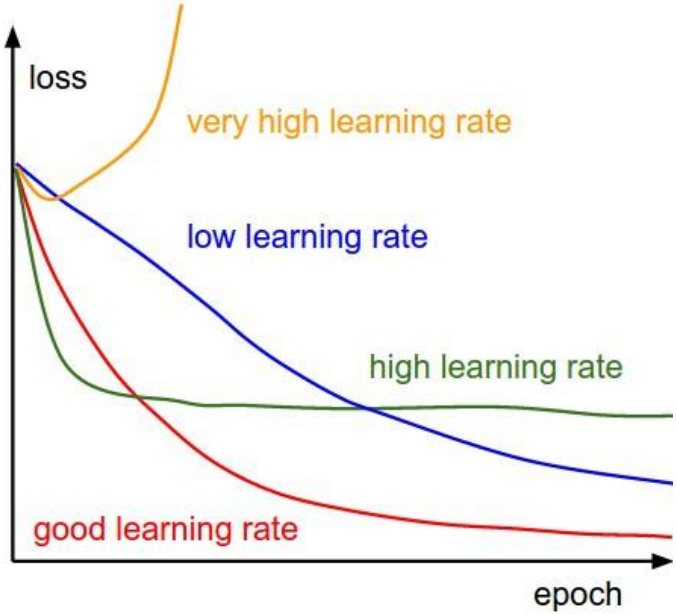
Effects of Initialization of weights

- Good initialization: Converge fast

Fine Points

- When do we stop the iterations?
  - When the gradient value is too low($< \varepsilon$)
  - When the change in objective is too small
- How about the step size?
  - Ensure smooth convergence

Gradient Descent - Learning rate



http://cs231n.github.io/neural-networks-3/#baby

Batch Gradient Descent

Pseudo Code

```
model = initialization(...)
train_data = load_training_data()
error = 0
for i in 1...n:
    for X, y in train_data :
       prediction = predict(X, model)
       error += calculate_error(y,prediction)
    gradient  = differentiate(error)
    model = update_model(model, gradient)
```

Mini Batch Gradient Descent

Pseudo Code

```
model = initialization(...)
train_data = load_training_data()
Tb1, Tb2, Tb3, ……,Tbm =
split_training_batches(train_data)
error = 0
for i in 1...n:
    for Tb in Tb1, Tb2, Tb3, ……,Tbn :
        for X, y in Tb :
            predictions = predict(X, model)
            error += calculate_error(y,predictions)
        gradient  = differentiate(error)
        model = update_model(model, gradient)
```

Stochastic Gradient Descent

Pseudo Code

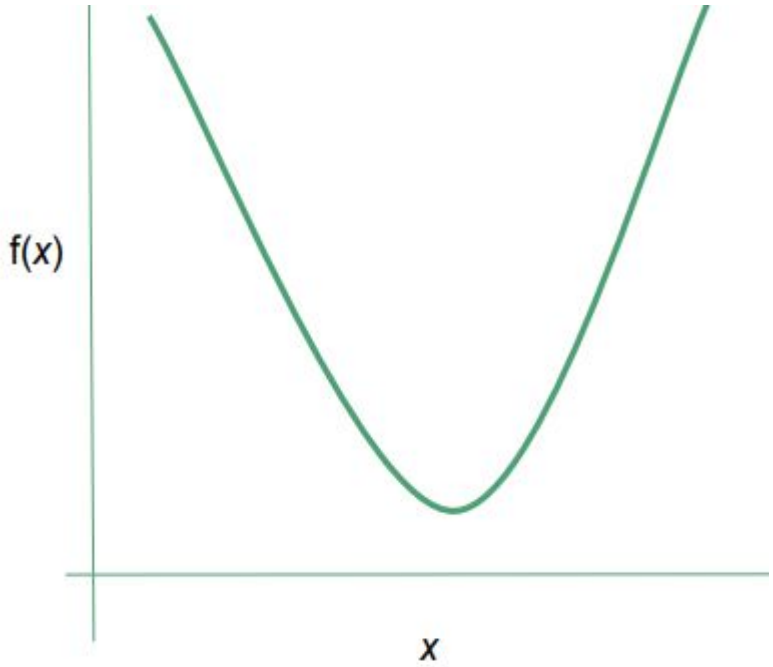
```
model = initialization(...)
train_data = load_training_data()
error = 0
for i in 1...n:
    for X, y in train_data :
       prediction = predict(X, model)
       error = calculate_error(y,prediction)
       gradient  = differentiate(error)
       model = update_model(model, gradient)
```

Trend

- Recent trend is to use Stochastic Mini Batch Gradient Descent

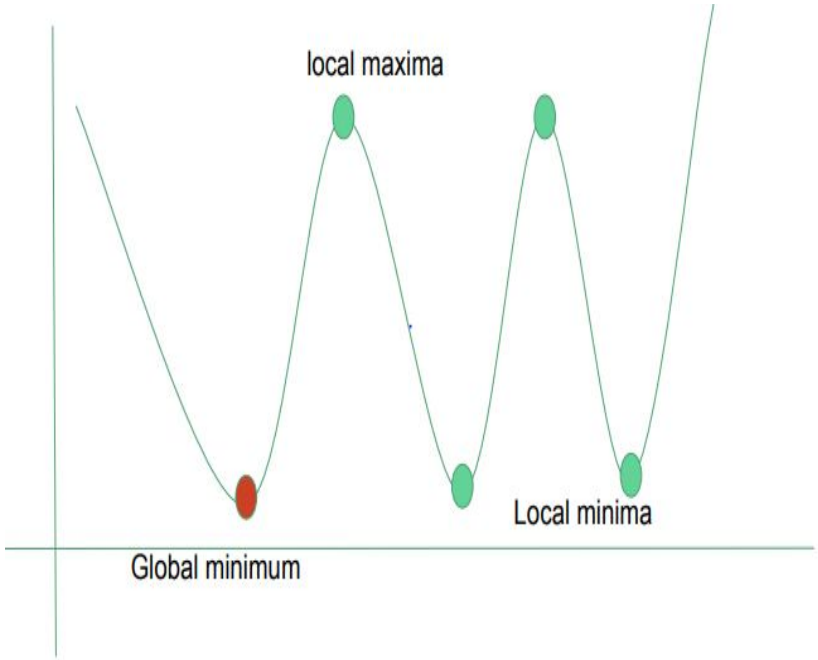
Convex Objective

- Local minima = global minima
- Strong theoretical guarantees
- Only one optimal solution, easier in intuition
- Takes lesser time to converge to optima



Non - Convex Objective

- Multiple local minima
- Multiple local minima != global minima
- it can take a lot of time to identify whether the problem has no solution or if the solution is global
- Takes longer to obtain global optima



Comments: Logistic Function

- The logistic function
  - Probability of an outcome

$$f(z) = \frac{1}{1 + e^{-z}}$$

- Has an interesting derivative form

$$f'(z) = f(z)(1 - f(z))$$

- Connect with linear:

$$z = w^T x \qquad f_w(z) = \frac{1}{1 + e^{-w^T x}}$$

Summary

- Linear classifiers are simple to train
- Gradient descent is a powerful technique to optimize convex and non-convex objective functions

Review: Pipeline/Concept Map

DATA

Structured (numerical, categorical attributes)

Digital Logs (Tweets, SMS)

RawData/Sensors (Image/Speech)

User behaviors

Etc.

FEATURE

Intuitive User defined

Raw data itself

Statistics (Histograms, PCA)

Signal Process (Fourier Xform)

FEATURE XFORMATIONS

Feature Selection

Feature Extraction

Dimensionality Reduction

Eg. PCA

ML PROBLEM

1. Classification
   a. Binary
   b. Multiclass
2. Regression
3. Clustering
4. Prediction (time series)

ALGORITHMS

1. KNN
2. Naïve Bayes
3. Perceptron
4. Linear
5. Decision Tree

PERFORM. METRICS

Accuracy

Confusion Matrix

Precision

Recall

AP

True Positive

Etc.

Thanks!

Questions?