MOD1

02 February 2025 09:52

Kernel machines

Kernel machines are a class of algorithms that leverage kernel functions to create non-linear decision boundaries allowing for complex data analysis

The main use of the kernel is to transforming data into high dimensions,

These are mostly effective in both classification as well as regression

Definition of Kernel function

→ A kernel function $K(x, y)$, calculates the inner product of 2 points x and y in a high-dimensional feature

common forms

- Linear Kernel
- Polynomial Kernel
- Radial Basis Function (RBF)
- Sigmoid Kernel

-) sigmoid kernel

Properties of Kernels

-) Symmetry

-) positive semidefiniteness

-) Inner product representation

* Roles of Kernels in non linear data

Kernel transform non-linear separable data into higher dimensional space.

-) linear kernel — $K(x, y) = x \cdot y$

-) polynomial $K(x, y) = (x \cdot y + c)^q$

-) RBF $K(x, y) = e^{-\gamma \|x - y\|^2}$

-) Sigmoid Kernel $K(x, y) = \tanh(x \cdot y + c)$

Intro to Kernels for Structured data & text

-) Kernel function compute similarities between data point in high-dimensional space

Point in high - dimensional space

→ Handling structured and unstructured data

→ NLP, Bioinformatics, social networks.

Structured Data

→ Data with predefined schema

→ trees, graphs

→ Cons:- finding relationship

Text

→ unstructured data, represented as sequence of word & characters

→ Articles, emails, and social media posts

→ Cons:- High dimensionality and sparsity.

Examples

1. Document classification
2. sentiment Analysis
3. spam detection

Q Biological sequence

why structured Data need special Kernels?

→ traditional Kernel fail to capture hierarchical & relation ship structure

why text Data need special Kernels?

→ High Dimensionality

→ Sparsity : most word or tokens in corpus appear infrequent

→ semantic meaning :- words with similar meaning may not share the same representation

→ order sensitivity :- word order impact meaning, require kernels that account for sequence

→ Heterogenous Data : text data can vary significantly in length and structure.

String Kernels :- measure the similarity based on common substring

TF-IDF Kernel :- Based on term frequency-inverse

Word Embedding Kernel : use embedding like word2vec or glove for similarity

Multiple Kernel learning

-) combines multiple kernel functions to improve predictions

→ Help integrate different data type

→ used in bioinformatics, computer vision, and NLP

Why multiple Kernels

→ A single kernel may not capture data complexity

→ Different datatypes need different kernels

→ improves accuracy

core concepts :-

Kernel functions :- Measure similarity between data points.

linear combination : uses a weighted sum of multiple kernels

optimization: learns the best kernel mix and classifier parameters.

input: taking multiple datatypes

Apply different kernel function

Combine kernels using weights

optimize weights and classifier

Final predictive model.

Linear Kernel :- capture direct relationship

RBF Kernel : Handles nonlinear data

Polynomial Kernel :- Models complex patterns

Pros & cons :

→ capture more data feature / types

-) improves prediction accuracy

-) works with multiple data sources

→ High computional cost

→ Complex optimizations

→ Complex optimization

→ Risk of overfitting with too many kernels

Emerging trends in MKL

→ Sparse MKL : uses only the most relevant kernels

→ Hierachical MKL : layered kernel approach for complex data

→ Deep Kernel learning : combines MKL with deep learning

Applications

→ Bio informatics

→ computer vision

→ NLP

→ Health care Analytics

Case Study Exam

Sentiment analysis using customer rev.

RBF Kernel :- used for numerical data

String Kernel : used for text review

ML combination: improve accuracy by leveraging both data sets

MOD2

02 February 2025 11:20

A real-time system processes a data stream. The first three observations are as follows:

Point	x_1	x_2	Label (y)
A	2	3	+1
B	-1	-2	-1
C	0	1	+1

$\eta = 0.4 \quad \lambda = 0.1$

$y_i (w x_i + b) \geq 1$

update weight

$\leftarrow w + \eta (y_i x_i - \lambda w)$

$[2, 3] \quad y = +1$

$= 1([0.0] \cdot [2, 3] + 0) = 0$

$= 0$

$a = y_i x_i - \lambda w$

$a = 1 \cdot [2, 3] - 0.1 [0, 0]$

$= [2, 3]$

$w + \eta (a)$

$= [0, 0] + 0.4 [2, 3]$

$w = [0.2, 0.3]$

$= (0.2, 0.3)$

$b = b + \eta y$

$b = 0 + 0.1 \; 1$

$= 0.1$

for iter 2

$x_1 = [-1, -2] \quad y = 1$

$y_i (w \cdot x + b)$

$1 ([0.2, 0.3][-1, -2] + 0.1)$

$-1 (0.9 + 0.1)$

$= -1 (-0.5)$

$= 0.7 < 1$

$w = w + \eta (y_i x - \lambda w)$

$a = y_i x - \lambda w$

$= -1[-1, -2] - 0.1[0.2, 0]$

$= (1, 2) - (0.01, 0.03)$

$a = (0.99, \; 1.97)$

$w = w + \eta (a)$

$= [0.2, 0.3] + 0.1 [0.99, 1.99]$

$= [0.2, 0.3] + [0.09, 0.19]$

$w = [0.29, 0.49]$

$b = b + \eta y$

$b = 0.1 + 0.1 (-1)$

$= 0.1 - 0.1$

$b = 0$

$x_3 = [0, 1 \quad] \quad y_3 = +1)$

$y.(wx+b) < 1$

$1([0.3, 0.5)(0, 1] + 0]$

$0.5 < 1$

$w = w + \eta(\alpha)$

$\alpha = yx \rightarrow w$

$= 1(0, 1) - 0.1(0.3, 0.5)$

$= (0, 1) - (0.03, 0.05)$

$= (0, 0.95)$

$w = [0.3, 0.5] + 0.1(0, 0.95)$

$\boxed{w = [0.3, 0.595)}$

$y.(wx+b) > 1$

$1([0.3, 0.6)[2, 3] + 0] > 1$

final weights $[0.3, 0.6]$

$0.3x_1 + 0.6x_2 = 0$

$\boxed{x_1 + 2x_2 = 0}$ Hyperplane

Margin $\frac{2}{\|w\|}$

$\|w\| = \sqrt{0.09 + 0.36}$

$= \sqrt{0.44}$

$= 0.67$

$= \frac{2}{0.67} = 2.98$

Theory

Hard margin SVM

→ to find a hyperplane that perfectly separate the classes in a dataset

→ used when data is linearly separable with out overlap

types

→ one type

zero misclassification

working

→ The svm tries to find hyperplane that maximize the margin

→ since misclassification is not allowed data points must classified correctly

→ It uses some constraints that no point can be within the margin or mis classified.

Advantage

→ Works well the data is perfectly separable

→ maximizes the margin

Disadvantage

→ Fails when data has noise or overlap

→ sensitive to outliers and misclassifications

Soft margin SVM

Goal

→ To allow some misclassifications inorder to create a more generable model

→ used when data is not perfectly separable

types

→ linear soft margin : Allow some misclassifications while maintaining a linear decision boundary

→ non-linear soft margin SVM :- use kernel to handle complex data distributions

working

→ Introduces slack variables (ξ) to allow some misclassified points

→ Balance between maximizing margin and minimizing classification error

→ uses a regularization parameter C to control the the trade off between a wider margin & few misclassifications

Advantages :-

→ Handles noisy and overlapping data better

→ more flexible in real world applications

→ can work with non linear data using kernel

Disadvantage :-

→ needs careful C parameter

→ computional expensive

Feature	Hard Margin SVM	Soft Margin SVM
Misclassification	Not allowed (strict)	Allowed (with penalties)
Data Type	Works only for perfectly separable data	Works for overlapping or noisy data
Slack Variables (ξ)	Not used	Used to allow margin violations
Outlier Sensitivity	Highly sensitive	More robust
Generalization	Poor for real-world data	Better generalization
Computational Cost	Lower	Higher (due to slack variables and optimization)

Distributed SVM

→ handles the data or large scale data that can't fit into memory of single machine. It used when data has to spread across diff locations

Challenges

→ Data partitioning & communication overheads

→ Scalability

→ Data imbalance

Algorithms

→ Parallel SVM

→ Mapreduce svm

→ Gradient Based distributed

workflow

→ Data partitioning : Divide the data across the multiple machines

→ Parallel training

→ Model Aggregation

→ model evaluation

→ Model update

Hyperparameter tuning

→ Regularization parameter C :

→ ctrl trade off b/w Min & Max misclassification

→ Kernel parameters :

→ learning rate

→ communication setting : tune communication intervals b/w nodes. to minimize overhead

Applications

-> Large Scale ing classfication

-> Big Data Analytics

-> Text mining

-> Bio informatics

-> Autonomous System

Online SVM

-> handles the streaming data, unlike regular SVM, where entire dataset to be stored in memory for training online SVM update the model incrementally as new data points arrive, making them suitable for situations

Challenges

-> Memory Mgmt

-> Model Drift

-> Computational Complexity

-> Model convergence

-> Hyperparameter tuning

Workflow

→ Data streaming

→ Incremental training

→ Model update

→ Model evaluation

→ Drift detection

tuning

→ Reg Parameter C

→ learning rate η

→ Kernel parameters

→ update frequency

Applications of online SVM

→ Real-time classification

→ Predictive maintenance

→ Spam filtering

→ Social media analytics

→ Sensor networks

MOD3

02 February 2025 11:20

Gradient Boosting / Ada Boosting