


Kernels II

Nico Görnitz
Technische Universität Berlin - Machine Learning Group
Beginner's Workshop Machine Learning 2018

Agenda

Methods

- Kernel Ridge Regression
- Kernel PCA
- One-class SVM and SVDD

Kernels for specific Tasks

- Basic Kernels re-visited
- Kernels for Sequences
- Kernels for Graphs and Trees
- Kernels for Probabilistic Models

Learning Kernels

- Multiple Kernel Learning

Kernel Approximations

- Random Fourier Features

Why use kernels?

1. Efficient computation in high-dimensional feature spaces
2. Non-linear feature maps for complex decision surfaces
3. Abstraction from data representation and learning methods

Agenda

Methods

Kernel Ridge Regression

Kernel PCA

One-class SVM and SVDD

Kernels for specific Tasks

Basic Kernels re-visited

Kernels for Sequences

Kernels for Graphs and Trees

Kernels for Probabilistic Models

Learning Kernels

Multiple Kernel Learning

Kernel Approximations

Random Fourier Features

Agenda

Methods

Kernel Ridge Regression

Kernel PCA

One-class SVM and SVDD

Kernels for specific Tasks

Basic Kernels re-visited

Kernels for Sequences

Kernels for Graphs and Trees

Kernels for Probabilistic Models

Learning Kernels

Multiple Kernel Learning

Kernel Approximations

Random Fourier Features

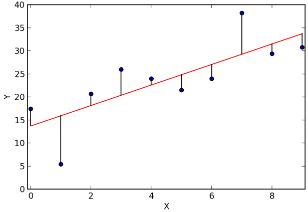
From OLS to Ridge Regression

The aim is to find a parameter vector $w_1 \in \mathbb{R}^d$ of a linear model $f_1(x) = b = \langle w_1, x \rangle$ that fits a given sample set $(x_i, b_i) \in \mathbb{R}^d \times \mathbb{R} \quad \forall i$ best. Hence, we are interested in solving the following least squares problem:

$$\min_{w \in \mathbb{R}^d} \sum_i (b_i - \langle w, x_i \rangle)^2$$

We introduce a regularization term $||w||^2$ into the optimization problem and a corresponding hyper-parameter $\lambda \geq 0$:

$$\min_{w \in \mathbb{R}^d} \mathcal{L}(w) = \min_{w \in \mathbb{R}^d} \lambda ||w||_2^2 + \sum_i (b_i - \langle w, x_i \rangle)^2$$



Solving the Ridge Regression Problem

For convenience we rephrase the latter into matrix notation.

$$\begin{aligned}\mathcal{L}(w) =& \lambda \|w\|^2 + \sum_i (b_i - \langle w, x_i \rangle)^2 \\ =& \lambda w^\mathsf{T} w + b^\mathsf{T} b - 2w^\mathsf{T} X b + w^\mathsf{T} X X^\mathsf{T} w\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{L}(w)}{\partial w} ! = 0 \Rightarrow 0 = 2\lambda w - 2Xb + 2XX^\mathsf{T} w \\ Xb = \lambda w + XX^\mathsf{T} w = (\lambda I + XX^\mathsf{T}) w \\ w = (\lambda I + XX^\mathsf{T})^{-1} Xb\end{aligned}$$

Kernel Ridge Regression Problem

We transform the data points into a (possibly very high dimensional) feature space using the feature mapping function $\phi : \mathbb{R}^d \to \mathcal{F}$. The resulting model, $f_2(x) = y = \langle w_2, \phi(x) \rangle$ with $w_2 \in \mathcal{F}$, retains the desired simplicity of linear functions while at the same time becoming much more expressive. The corresponding optimization problem arrives at:

$$\min_{w \in \mathcal{F}} \mathcal{L}(w) = \min_{w \in \mathcal{F}} \lambda \|w\|_2^2 + \sum_i \left(b_i - \langle w, \phi(x_i) \rangle\right)^2$$

which reads in matrix notation $\mathcal{L}(w) = \lambda w^\mathsf{T} w + b^\mathsf{T} b - 2 w^\mathsf{T} \Phi b + w^\mathsf{T} \Phi \Phi^\mathsf{T} w$. We can attempt to solve it the same way as ridge regression:

$$\frac{\partial \mathcal{L}(w)}{\partial w} ! = 0 \;\Rightarrow\; w = \left(\lambda I + \Phi \Phi^\mathsf{T}\right)^{-1} \Phi b \,,$$

which, unfortunately, does not help (i.e. $\Phi \Phi^\mathsf{T}$ is a covariance matrix in the possibly very high dimensional feature space!).

Kernel Ridge Regression

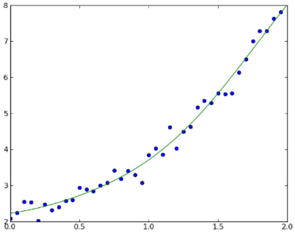
Now, we make use of a special case of the **Woodbury identity** for positive definite matrices P and R:

$$(P^{-1} + B^{\mathsf{T}} R^{-1} B)^{-1} B^{\mathsf{T}} R^{-1} = PB^{\mathsf{T}} (BPB^{\mathsf{T}} + R)^{-1}$$

In our problem, $R^{-1} = R = I$, $B^{\mathsf{T}} = \Phi$ and $P^{-1} = \lambda I$:

$$w = (\lambda I + \Phi\Phi^{\mathsf{T}})^{-1} \Phi b = \frac{1}{\lambda}\Phi(\frac{1}{\lambda}\Phi^{\mathsf{T}}\Phi + I)^{-1} b$$

$$w = \Phi(\Phi^{\mathsf{T}}\Phi + \lambda I)^{-1} b$$



which can be rephrased as $w = \sum_i \alpha_i \phi(x_i)$
with $\alpha = (\Phi^{\mathsf{T}}\Phi + \lambda I)^{-1} b = (K + \lambda I)^{-1} b$.

Agenda

Methods

- Kernel Ridge Regression
- Kernel PCA
- One-class SVM and SVDD

Kernels for specific Tasks

- Basic Kernels re-visited
- Kernels for Sequences
- Kernels for Graphs and Trees
- Kernels for Probabilistic Models

Learning Kernels

- Multiple Kernel Learning

Kernel Approximations

- Random Fourier Features

Principle Components Analysis

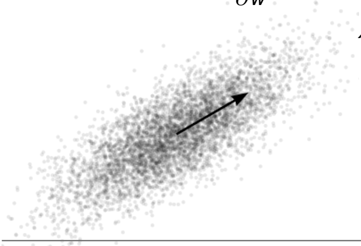
Find the direction of maximum variance w given (centered!) datapoints $x_1, \ldots, x_n \in \mathbb{R}^d$:

$$\max_w \quad \sum_i (w^\intercal x_i)^2 = w^\intercal X X^\intercal w \quad \text{subject to} \quad \|w\|^2 \leq 1\,.$$

Solve the corresponding Lagrangian $\mathcal{L}(w, \lambda) = w^\intercal X X^\intercal w - \lambda(w^\intercal w - 1)$ for w:

$$\frac{\partial \mathcal{L}(w, \lambda)}{\partial w} ! = 0 \;\Rightarrow\; 0 = 2XX^\intercal w - 2\lambda w$$

$$XX^\intercal w = \lambda w \quad \text{(=Eigenwert problem)}$$



Kernel Principle Components Analysis

From the unconstrained optimization problem (literally that's how we designed the OP), we know that the optimal solution of w will lie in the span of the data $w = \sum_i \alpha_i x_i = X\alpha$. Now lets do the feature map trick again:

$$\max_w \quad \sum_i (w^\mathsf{T}\phi(x_i))^2 = w^\mathsf{T}\Phi\Phi^\mathsf{T} w \quad \text{subject to} \quad \|w\|^2 \leq 1\,.$$

Solving is similar to standard PCA, i.e. the solution remains $\Phi\Phi^\mathsf{T} w = \lambda w$. Now, let's extend the solution:

$$\begin{aligned}
\Phi\Phi^\mathsf{T} w &= \lambda w \quad &&|\text{substitute } w = \Phi\alpha \\
\Phi\Phi^\mathsf{T}\Phi\alpha &= \lambda\Phi\alpha \quad &&|\cdot \Phi^\mathsf{T} \\
\Phi^\mathsf{T}\Phi\Phi^\mathsf{T}\Phi\alpha &= \lambda\Phi^\mathsf{T}\Phi\alpha \quad &&|\text{substitute } K = \Phi^\mathsf{T}\Phi \\
K^2\alpha &= \lambda K\alpha \quad &&\Rightarrow K\alpha = \lambda\alpha
\end{aligned}$$

Centering in feature space is important!

Agenda

Methods

Kernel Ridge Regression

Kernel PCA

One-class SVM and SVDD

Kernels for specific Tasks

Basic Kernels re-visited

Kernels for Sequences

Kernels for Graphs and Trees

Kernels for Probabilistic Models

Learning Kernels

Multiple Kernel Learning

Kernel Approximations

Random Fourier Features

What is One-class Classification?

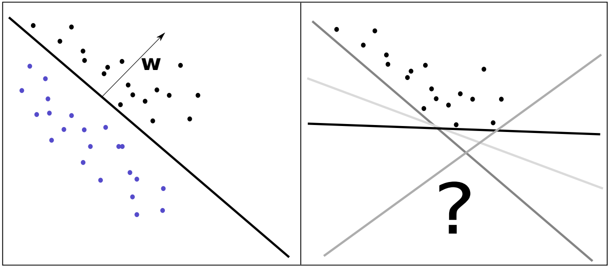
Learn common properties of given examples and be able to tell if a test point has the same properties or not:

- Assuming we know the data distribution $p(x)$ of our data. The task is, to reject all data points with $p(x) < \nu$ given a pre-defined threshold ν.
- Unfortunately, we usually don't know $p(\cdot)$ and estimation is really hard ...
- Therefore, we estimate a function $f(x) \in \{+1, -1\}$ that tells us whether $p(x) < \nu$ or not

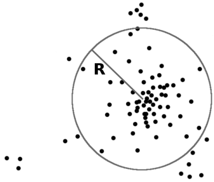
A Machine Learning Approach...

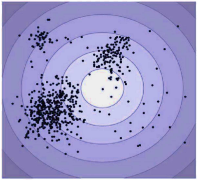
– Convert measurements to vector space and use famous SVM. Easy! Isn't it?

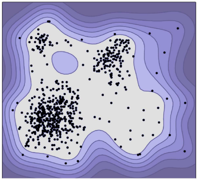
– No! (a) We have no ground truth, (b) most data points exhibit normal behavior, (c) new classes of object might occur during application



Support Vector Data Description (SVDD)





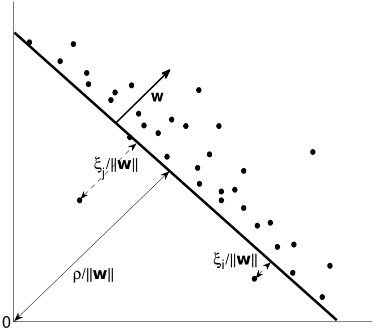


Support Vector Data Description (SVDD)

- Compute minimal enclosing sphere with center c and radius R
- Anomaly score as the distance to center c, that is $f(x) = \|\phi(x) - c\|$
- Accept data point x if $f(x) \leq R$ and ...

  ...reject x if $f(x) > R$

One-class Support Vector Machines (OC-SVM)



One-class SVM

- Separate data from origin with hyperplane with maximum distance to origin
- Model function: $f(x) = \langle w, \phi(x) \rangle - \rho$

OC-SVM: Optimization Problem

Primal optimization problem $0 < \nu \leq 1$

$$\min_{w,\rho,\xi} \quad \frac{1}{2}\|w\|^2 - \rho + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i$$

$$\text{s.t.} \quad \forall_{i=1}^{n} : \langle w, \phi(x_i)\rangle \geq \rho - \xi_i \quad \text{and} \quad \xi_i \geq 0$$

Lagrange: $\mathcal{L} = \frac{1}{2}\|w\|^2 - \rho + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i + \sum_i \alpha_i\left(\rho - \xi_i - \langle w, \phi(x_i)\rangle\right) - \sum_i \beta_i\xi_i$ with $\alpha_i, \beta_i \geq 0$.

$$\frac{\partial\mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i}^{n}\alpha_i\phi(x_i)$$

$$\frac{\partial\mathcal{L}}{\partial \xi_i} = 0 \Rightarrow \frac{1}{n\nu} - \alpha_i - \beta_i = 0 \Rightarrow 0 \leq \alpha_i \leq \frac{1}{n\nu}$$

$$\frac{\partial\mathcal{L}}{\partial \rho} = 0 \Rightarrow 1 = \sum_{i}^{n}\alpha_i$$

OC-SVM: Optimization Problem

Primal optimization problem $0 < \nu \leq 1$

$$\min_{w,\rho,\xi} \quad \frac{1}{2}\|w\|^2 - \rho + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i$$

$$\text{s.t.} \quad \forall_{i=1}^{n} : \langle w, \phi(x_i)\rangle \geq \rho - \xi_i \quad \text{and} \quad \xi_i \geq 0$$

Dual optimization problem

$$\max_{\alpha} \quad -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j K(x_i, x_j)$$

$$\text{s.t.} \quad \sum_{i=1}^{n}\alpha_i = 1 \quad \text{and} \quad 0 \leq \alpha_i \leq \frac{1}{n\nu} \quad \forall i$$

And expansion $w = \sum_i^n \alpha_i \phi(x_i)$

Question: What happens for $\nu = 1$?

Agenda

Methods

- Kernel Ridge Regression
- Kernel PCA
- One-class SVM and SVDD

Kernels for specific Tasks

- Basic Kernels re-visited
- Kernels for Sequences
- Kernels for Graphs and Trees
- Kernels for Probabilistic Models

Learning Kernels

- Multiple Kernel Learning

Kernel Approximations

- Random Fourier Features

Agenda

Methods

- Kernel Ridge Regression
- Kernel PCA
- One-class SVM and SVDD

Kernels for specific Tasks

- Basic Kernels re-visited
- Kernels for Sequences
- Kernels for Graphs and Trees
- Kernels for Probabilistic Models

Learning Kernels

- Multiple Kernel Learning

Kernel Approximations

- Random Fourier Features

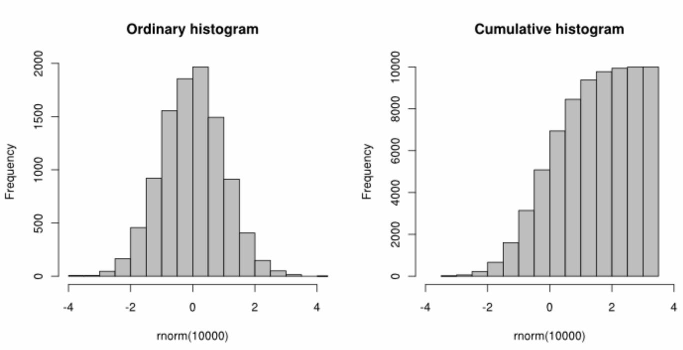
Examples

- Exponential and Laplacian Kernel
- Hyperbolic Tangent (Sigmoid) Kernel
- (Inverse) Multiquadric Kernel
- Circular and Spherical Kernel
- Power Kernel (Sahbi and Fleuret, 2004)
- Log Kernel
- Spline Kernel (Gunn, 1998)
- Bessel Kernel
- Cauchy Kernel (Basag, 2008)
- Chi-Square Kernel (Vedaldi and Zisserman, 2011)
- Wavelet kernel (Zhang et al, 2004)
- **Histogram Intersection Kernels** (Barla et al., ICIP 2003)

Histogram Intersection Kernel (Barla et al., ICIP 2003)

Definition (Histogram)

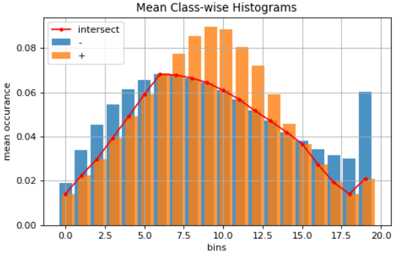
A histogram is an accurate representation of the distribution of numerical data. It is an estimate of the probability distribution of a continuous variable.

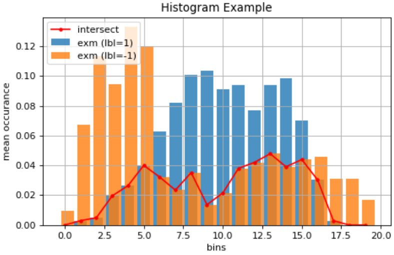


Histogram Intersection Kernel (Barla et al., ICIP 2003)

Assume data points are histograms consisting of B bins each. Then the histogram intersection kernel K is defined as:

$$K(x, y) = \sum_{b=1}^{B} \min(x_b, y_b) = \frac{1}{2} \sum_{b=1}^{B} (x_b + y_b - |x_b - y_b|)$$





Agenda

Methods

- Kernel Ridge Regression
- Kernel PCA
- One-class SVM and SVDD

Kernels for specific Tasks

- Basic Kernels re-visited
- Kernels for Sequences
- Kernels for Graphs and Trees
- Kernels for Probabilistic Models

Learning Kernels

- Multiple Kernel Learning

Kernel Approximations

- Random Fourier Features

Examples

- Weighted Degree Kernel (Rätsch et al., Bioinformatics, 2005)
- Spectrum Kernel (Leslie et al., Pac. Symp. Biocomputing 2002)
- **Bag-of-words Kernel**

What are sequences?

Alphabet

An alphabet $\mathcal{A}$ is a finite set of discrete symbols.

- DNA, $\mathcal{A} = \{A, C, G, T\}$
- Natural language text, $\mathcal{A} = \{a, b, c, \ldots, A, B, C, \ldots\}$

Sequence

A sequence x is concatenation of symbols from $\mathcal{A}$, i.e. $x \in \mathcal{A}^*$.

- $\mathcal{A}^n$ is the set of all sequences of length n
- $\mathcal{A}^*$ is the set of all sequences of arbitrary length
- $|x|$ is the length of sequence x

Embedding Sequences

Characterize sequences using a language $L \subset \mathcal{A}^*$

Feature space spanned by frequencies of words $w \in L$

Feature Map

A function $\phi : \mathcal{A}^* \to \mathbb{R}^{|L|}$ mapping sequences to $\mathbb{R}^{|L|}$ given by

$$x \mapsto \left(\#_w(x)\sqrt{N_w}\right)_{w \in L}$$

where $\#_w(x)$ returns the frequency of w in sequence x.

Refinement of embedding using weighting constants N_w

Normalization, often $\|\phi(x)\|_1 = 1$ or $\|\phi(x)\|_2 = 1$

Generic Sequence Kernel

Generic Sequence Kernel

A sequence kernel $K : \mathcal{A}^* \times \mathcal{A}^* \to \mathbb{R}$ over ϕ is defined by

$$K(x, y) = \langle \phi(x), \phi(y) \rangle = \sum_{w \in L} \#_w(x) \#_w(y) N_w \ .$$

By definition K is an inner product in $\mathbb{R}^{|L|}$ and thus symmetric and positive semi-definite.

Feature space induced by ϕ explicit but sparse.

Bag-of-Words Kernel

Characterization of sequences by non-overlapping words.

$$x = \text{"Hasta la vista, baby."} \longrightarrow \{\text{"Hasta", "la", "vista", "baby"}\}$$

Bag-of-Words Kernel

Sequence kernel using embedding language containing words

$$L = \text{Dictionary (explicit) or } L = (A \setminus D)^* \text{ (implicit)}$$

with $D \subset A$ delimiter symbols, e.g. punctation and space.

Extension using stemming techniques, "helping" $\Rightarrow$ "help"

Weighting to control contribution of words

Bag-of-Words: Example

Bag of Words Example

Document 1

The quick brown fox jumped over the lazy dog's back.

Document 2

Now is the time for all good men to come to the aid of their party.

Term	Document 1	Document 2
aid	0	1
all	0	1
back	1	0
brown	1	0
come	0	1
dog	1	0
fox	1	0
good	0	1
jump	1	0
lazy	1	0
men	0	1
now	0	1
over	1	0
party	0	1
quick	1	0
their	0	1
time	0	1

Stopword List

for
is
of
the
to

Agenda

Methods

- Kernel Ridge Regression
- Kernel PCA
- One-class SVM and SVDD

Kernels for specific Tasks

- Basic Kernels re-visited
- Kernels for Sequences
- Kernels for Graphs and Trees
- Kernels for Probabilistic Models

Learning Kernels

- Multiple Kernel Learning

Kernel Approximations

- Random Fourier Features

Examples

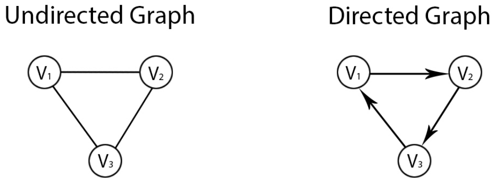
- Diffusion Kernel (Kondor and Lafferty, 2002)
- Approximate tree kernels (Rieck et al., JMLR 2010)
- Graphlet Kernel (Borgwardt, Petri, et al., MLG 2007)
- Cyclic Pattern Kernel (Horvath et al., KDD 2004)
- Subtree Kernel (Ramon and Gaertner, 2004)
- Edit-Distance Kernel (Neuhaus and Bunke, 2006)
- Weighted Decomposition Kernel (Menchetti et al., ICML 2005)
- Optimal Assignment Kernel (Froehlich et al., ICML 2005)
- Shortest-Path Kernel (Borgwardt and Kriegel, ICDM 2005)
- **Random Walks Kernel** (Kashima et al., ICML 2003, Gaertner et al., COLT 2003)

Technische Universität Berlin

What is a graph?

Definition (Graph)

A graph G is an ordered pair $G = (V, E)$ comprising a set $V = (v_i)_{i=1,\dots,n}$ of n vertices together with a set $E \subset V \times V$ of edges (which are 2-element subsets of V.



e.g. $G_{undirected} = (V, E)$ with $E = \{(v_1, v_2), (v_1, v_3), (v_2, v_3)\}$ and $V = \{v_1, v_2, v_3\}$. Such graphs can be represented by their respective adjacency matrix $A \in \{0, 1\}^{|V| \times |V|}$ with $a_i j = 1$ if $(v_i, v_j) \in E$.

Graph Comparison Problem

Definition (Graph Comparison Problem)

Given two graphs G and G' from the space of graphs $\mathcal{G}$. The problem of graph comparison is to find a mapping

$$s : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$$

such that $s(G, G')$ quantifies the (dis)similiarity between G and G'.

Important for, e.g.

- Function prediction of chemical compounds
- Structural comparison and function prediction of protein structures
- Comparison of social networks
- Analysis of semantic structures in Natural Language Processing
- Comparison of UML diagrams

Solutions I: Subgraph Isomorphism

Principle

Graph Isomorphism

Find a mapping f of the vertices of G_1 to the vertices of G_2 such that G_1 and G_2 are identical; i.e. (x, y) is an edge of G_1 iff $(f(x), f(y))$ is an edge of G_2. Then f is an isomorphism, and G_1 and G_2 are called isomorphic.

Subgraph Isomorphism

Subgraph isomorphism asks if there is a subset of edges and vertices of G_1 that is isomorphic to a smaller graph G_2.

Advantages

– Captures topological similarities between graphs accurately

Disadvantages

– Runtime may grow exponentially with the number of nodes

Solutions II: Graph Edit Distances

Principle

- Count operations that are necessary to transform G_1 into G_2
- Assign costs to different types of operations (edge/node insertion/deletion, modification of labels)

Advantages

- Captures partial similarities between graphs
- Allows for noise in the nodes, edges and their labels
- Flexible way of assigning costs to different operations

Disadvantages

- Contains subgraph isomorphism check as one intermediate step
- Choosing cost function for different operations is difficult

Solutions III: Topological Descriptors

Principle

- Map each graph to a feature vector
- Use distances and metrics on vectors for learning on graphs

Advantages

- Reuses known and efficient tools for feature vectors

Disadvantages

- Efficiency comes at a price: feature vector transformation leads to loss of topological information

Example: Random Walks Kernel

Principle

- Count common walks in two input graphs G and G'
- Walks are sequences of nodes that allow repetitions of nodes

Elegant computation

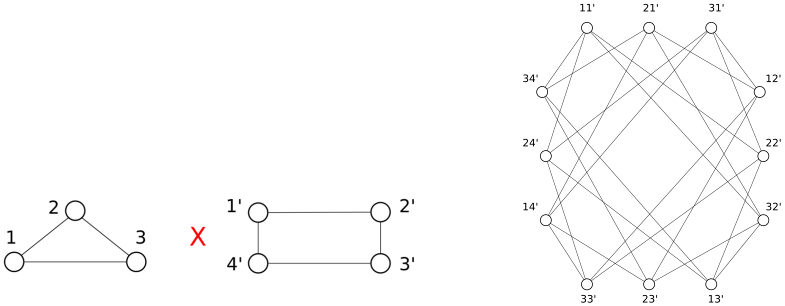
- Walks of length k can be computed using the k-th power of the adjacency matrix $A \in \mathbb{R}^{n \times n}$
- Construct direct product graph of G and G'
- Count walks in this product graph $G_x = (V_x, E_x)$
- Each walk in the product graph corresponds to one walk in G and G'

$$K(G, G') = \sum_{i,j=1}^{|V_x|} [\sum_k \lambda^k A_x^k]_{ij}$$

Example: Direct product of two graphs

Definition (Direct Product Graph)

Given two graphs $G = (V, E)$ and $G' = (V', E')$, their direct product G_x is a new graph with $V_x = \{(v_i, v_r') : v_i \in V, v_r' \in V'\}$ and $E_x = \{((v_i, v_r'), (v_j, v_s')) : (v_i, v_j) \in E \wedge (v_r', v_s') \in E'\}$.



[Vishwanathan et al., JMLR, 2010]

Technische Universität Berlin

Agenda

Methods

- Kernel Ridge Regression
- Kernel PCA
- One-class SVM and SVDD

Kernels for specific Tasks

- Basic Kernels re-visited
- Kernels for Sequences
- Kernels for Graphs and Trees
- Kernels for Probabilistic Models

Learning Kernels

- Multiple Kernel Learning

Kernel Approximations

- Random Fourier Features

Examples

- Probability Product Kernels (Jebara et al., JMLR, 2004)
- Bhattacharyya kernel (Jebara et al., JMLR, 2004)
- Expected likelihood kernel (Jebara et al., JMLR, 2004)
- Bayesian Kernel (Alashwal et al., WOSET, 2009)
- TOP kernel (Tsuda et al., NIPS, 2002)
- **Fisher Kernel** (Jaakkola and Haussler, 1999)

The Fisher Kernel (Jaakkola and Haussler, 1999)

Suppose that we are given a probabilistic model of our data $p(x, \theta)$ which is parameterized by θ. Then the Fisher kernel for two datapoints x and x' is defined as the inner product (with special normalization) of the derivatives (with respect to the parameters θ) of the log-likelihoods $p(x|\theta)$ and $p(x'|\theta)$:

$$K(x, x') = s(x, \theta)^\intercal Z_\theta^{-1} s(x', \theta)$$

$$s(x, \theta) = \frac{\partial}{\partial \theta} \log p(x|\theta)$$

Z denotes the Fisher information matrix: $Z_\theta = \mathbb{E}_x[s(x, \theta)s(x, \theta)^\intercal |\theta]$

Practically, the Z_θ is replace by the identity matrix.

Agenda

Methods

- Kernel Ridge Regression
- Kernel PCA
- One-class SVM and SVDD

Kernels for specific Tasks

- Basic Kernels re-visited
- Kernels for Sequences
- Kernels for Graphs and Trees
- Kernels for Probabilistic Models

Learning Kernels

- Multiple Kernel Learning

Kernel Approximations

- Random Fourier Features

Recap: Support Vector Machine

Primal SVM:

$$\min_{\tilde{w},b,\xi} \quad \frac{1}{2}\|\tilde{w}\|^2 + C\sum_i \xi_i$$

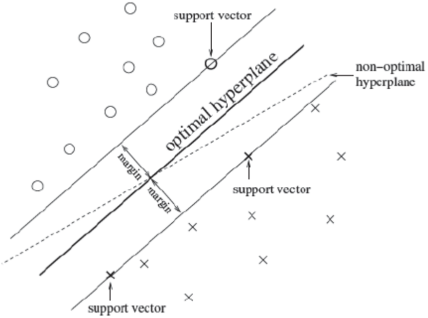
subject to $\quad y_i(\langle \tilde{w}, \phi(x_i)\rangle + b) \geq 1 - \xi_i$

$$\xi_i \geq 0, \quad i = 1, \ldots, n$$

Dual SVM:

$$\max_{\alpha} \quad \sum_i \alpha_i - \frac{1}{2}\sum_{ij} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

subject to $\quad \sum_i \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1$



Agenda

Methods

- Kernel Ridge Regression
- Kernel PCA
- One-class SVM and SVDD

Kernels for specific Tasks

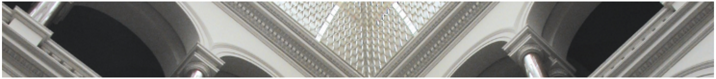
- Basic Kernels re-visited
- Kernels for Sequences
- Kernels for Graphs and Trees
- Kernels for Probabilistic Models

Learning Kernels

- Multiple Kernel Learning

Kernel Approximations

- Random Fourier Features



Technische Universität Berlin

Idea of MKL

Instead of selecting a single, best kernel, why not learn combinations of all suitable kernels?
Hence, in multiple kernel learning, we want to find a weighted combination of kernels $K = \sum_t d_t K_t$ (for $d_t \geq 0$ and $\|d\|_p^2 = 1$) that solves the problem best.

Problem: How to adjust weighting parameters d_t?

Solution:

1. A weighted kernel can be expressed as inner product of weighted feature maps:

$$K(x, y) = \sum_t d_t K_t(x, y) = \sum_t d_t \langle \phi_t(x), \phi_t(y) \rangle = \sum_t \langle \sqrt{d_t}\phi_t(x), \sqrt{d_t}\phi_t(y) \rangle$$

2. Weighting the parameters instead of the features:

$$\langle \tilde{w}_t, \sqrt{d_t}\phi_t(x) \rangle = \langle \sqrt{d_t}\tilde{w}_t, \phi_t(x) \rangle = \langle \sqrt{d_t}\tilde{w}_t, \phi_t(x) \rangle$$

3. Substituting $\sqrt{d_t}\tilde{w}_t = w_t$ and hence, $\tilde{w} = \frac{1}{\sqrt{d_t}} w_t$

Multiple Kernel Learning (SVM)

Primal MKL-SVM:

$$\min_{w,b,d,\xi} \quad \frac{1}{2}\sum_t d_t^{-1}\|w_t\|^2 + C\sum_i \xi_i$$

$$\text{subject to} \quad y_i\big(\sum_t \langle w_t, \phi_t(x_i)\rangle + b\big) \geq 1 - \xi_i$$

$$\xi_i \geq 0, \quad d_t \geq 0, \quad \|d\|_p^2 \leq 1, \quad i = 1, \ldots, n, \; t = 1, \ldots, T$$

Dual MKL-SVM:

$$\max_{\alpha} \quad \min_{d_t \geq 0, \|d\|_p^2 \leq 1} \quad \overbrace{\sum_i \alpha_i - \frac{1}{2}\sum_{ij}\alpha_i\alpha_j y_i y_j \sum_t d_t K_t(x_i, x_j)}^{J(\alpha,d)}$$

$$\text{subject to} \quad \sum_i \alpha_i y_i = 0 \quad 0 \leq \alpha_i \leq C \; i = 1, \ldots, n$$

Multiple Kernel Learning (SVM)

Algorithm 1 Algorithm (Multiple Kernel Learning for SVM)

Require: x, y, C, p-norm and a list of kernels K_t

Initialize kernel mixture coefficients such that $\|d^{z=0}\|_p = 1$

while Until Convergence **do**

(Step 1) solve standard SVM problem: $\alpha^{z+1} = \arg\max_{\alpha:0\leq\alpha_i\leq C} J(\alpha, d^z)$ s.t. $\sum_{i=1}^n \alpha_i y_i = 0$

(Step 2) optimize the kernel weights: $d^{z+1} = \arg\min_{d_t\geq 0} J(\alpha^{z+1}, d)$ s.t. $\|d\|_p^2 \leq 1$

z=z+1

end while

return Trained parameter vector $\alpha^\star$, weights d^*

Step 2 can be solved analytically: $d_t = \frac{\|w_t\|_2^{\frac{2}{p+1}}}{\left(\sum_{t'} \|w_{t'}\|_2^{\frac{2p}{p+1}}\right)^{\frac{1}{p}}}$ with expansions $w_t = d_t \sum_i \alpha_i y_i \phi(x_i)$.

Agenda

Methods

- Kernel Ridge Regression
- Kernel PCA
- One-class SVM and SVDD

Kernels for specific Tasks

- Basic Kernels re-visited
- Kernels for Sequences
- Kernels for Graphs and Trees
- Kernels for Probabilistic Models

Learning Kernels

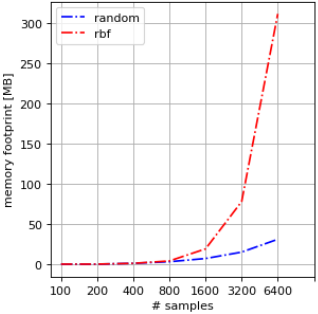
- Multiple Kernel Learning

Kernel Approximations

- Random Fourier Features

Why Approximations?

Although kernels are very versatile and often easier to construct than explicit feature maps, they quickly show their limits in large-scale and big data setting due to their quadratic scaling:



We would like to build a feature function given a specific choice of kernel such that $K(x, y) \approx \langle \phi(x), \phi(y) \rangle$.

Agenda

Methods

Kernel Ridge Regression

Kernel PCA

One-class SVM and SVDD

Kernels for specific Tasks

Basic Kernels re-visited

Kernels for Sequences

Kernels for Graphs and Trees

Kernels for Probabilistic Models

Learning Kernels

Multiple Kernel Learning

Kernel Approximations

Random Fourier Features

Approximating Kernels: Theory

Theorem (Bochner)

Assume that K is a normalized, positive definite, translation-invariant (real) kernel (i.e. $K(x, y) = K(x - y)$ and $K(0) = 1$) then it can be represented as the Fourier transform of a probability distribution:

$$
\begin{aligned}
K(\Delta) &= \int \exp(iw^T\Delta)p(w)dw \\
&= \mathbb{E}_{w\sim p}[\exp(iw^T\Delta)] = \mathbb{E}_{w\sim p}[\cos(w^T\Delta)] = \mathbb{E}_{w\sim p}[\cos(w^T(x-y))] \\
&= \mathbb{E}_{w\sim p}[\cos(w^Tx)\cos(w^Ty) + \sin(w^Tx)\sin(w^Ty)] \\
&= \mathbb{E}_{w\sim p}[[\cos(w^Tx), \sin(w^Tx)] \cdot [\cos(w^Ty), \sin(w^Ty)]^T]
\end{aligned}
$$

Hence, we can approximate K using D $\sin(w^Tx)$ and $\cos(w^Tx)$ features with randomly sampled $w \sim p$ and the empirical expectation:

$$K(x, y) \approx \frac{1}{D}\sum_d [\cos(w_d^Tx), \sin(w_d^Tx)] \cdot [\cos(w_d^Ty), \sin(w_d^Ty)]^T$$

Approximating Gaussian Kernels: Application

- If the kernel is Gaussian with unit variance $K(\Delta) = \exp(-\|\Delta\|^2/2)$ then the corresponding sampling distribution is $p(w) = (2\pi)^{-\frac{|\mathcal{X}|}{2}} \exp(-\|w\|^2/2)$
- Variant: The kernel is approximated by sampling only cos-features (instead of cos and sin) but with an additional shift $b \sim unif(0, 2\pi)$.

Algorithm 2 Algorithm (Random Fourier Features)

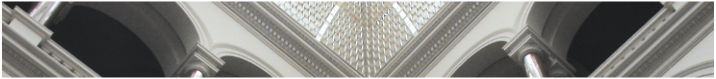
Require: Input data $x_i \in \mathcal{X}$, $i = 1, \ldots, n$, number of random features D

Sample $j = 1, \ldots, D$ offsets $b_j \sim unif(0, 2\pi)$

Sample $j = 1, \ldots, D$ $|\mathcal{X}|$-dimensional parameters $w_j \sim \mathcal{N}(0, 1)$

Construct the approximate feature map $\phi(x) = \sqrt{(\frac{2}{D})}[\cos(w_1^T x + b_1), \ldots, \cos(w_D^T x + b_D)]^T$

return transformed data $\phi(x_i) \in \mathbb{R}^D$, $i = 1, \ldots, n$



Thank you!