Kernels

Instructor: Jessica Wu -- Harvey Mudd College

The instructor gratefully acknowledges Andrew Ng (Stanford), Eric Eaton (UPenn), Tommi Jaakola (MIT) and the many others who made their course materials freely available online.

SVMs

Learning Goals

✓ Solving the SVM optimization problem
  – using Lagrange multipliers (leads to dual problem)

✓ Allowing misclassified examples
  – using slack variables (leads to soft-margin SVM)

✓ Describe the SVM loss function

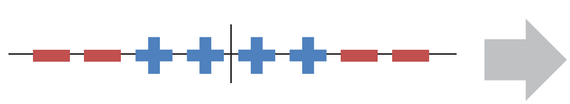
- Allowing non-linear decision boundaries
  – using kernels

Kernel Basics

Learning Goals

- Motivate kernels
  - mapping to new feature space
  - easy to use in SVM optimization and prediction
- Define kernels formally
  - what makes a kernel valid

When Linear Separators Fail

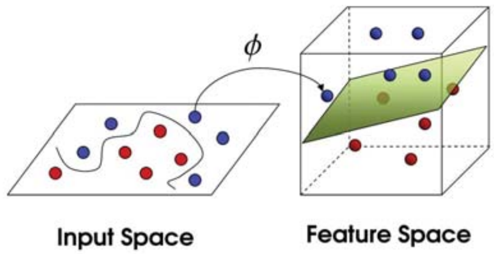


$x \in \mathbb{R}$

not linearly separable

Based on slide by Eric Eaton [originally by Tim Oates]

Mapping to a New Feature Space



$$\phi : \mathcal{X} \mapsto \mathcal{F}$$

Example: for $\boldsymbol{x} \in \mathbb{R}^2$,

$$\boldsymbol{\phi}([x_1,\, x_2]^T) = [x_1,\, x_2,\, x_1 x_2,\, {x_1}^2,\, {x_2}^2]^T$$

Rather than run SVM on $\boldsymbol{x}$, run it on $\boldsymbol{\phi}(\boldsymbol{x})$

Find non-linear separator in input space

Based on slide by Eric Eaton [originally by Tim Oates; image from http://web.engr.oregonstate.edu/ ~afern/classes/cs534/]

Using Mapped Features for SVM

Primal
$$\min_{\boldsymbol{\theta}, b} \frac{1}{2} \|\boldsymbol{\theta}\|^2$$
$$\text{s.t. } y^{(i)} \left(\boldsymbol{\theta}^T \boldsymbol{x}^{(i)} + b \right) \geq 1 \text{ for } i = 1, \ldots, n$$

Dual
$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y^{(i)} y^{(j)} \langle \boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)} \rangle$$
$$\text{s.t. } \alpha_i \geq 0 \qquad \text{for } i = 1, \ldots, n$$
$$\sum_{i=1}^{n} \alpha_i y^{(i)} = 0$$

Solution
$$\boldsymbol{\theta} = \sum_{i=1}^{n} \alpha_i y^{(i)} \boldsymbol{x}^{(i)}$$
for SV $x^{(i)}$, $\quad b = y^{(i)} - \boldsymbol{\theta}^T \boldsymbol{x}^{(i)}$

Prediction
$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} + b$$
$$= \left(\sum_{i=1}^{n} \alpha_i y^{(i)} \boldsymbol{x}^{(i)} \right)^T \boldsymbol{x} + b$$
$$= \sum_{i=1}^{n} \alpha_i y^{(i)} \langle \boldsymbol{x}^{(i)}, \boldsymbol{x} \rangle + b$$

If we have found optimal α_i's (from dual formulation), then to make a prediction for x, we only have to calculate a quantity dependent on dot product between x and SVs in training set.

Key Insight

Optimization

$$\left\langle \boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right), \boldsymbol{\phi}\left(\boldsymbol{x}^{(j)}\right)\right\rangle$$

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y^{(i)} y^{(j)} \langle \boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)} \rangle$$

$$\text{s.t. } \alpha_i \geq 0 \qquad \text{for } i = 1, \ldots, n$$

$$\sum_{i=1}^{n} \alpha_i y^{(i)} = 0$$

optimization depends only on inner products between inputs $\langle \boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)} \rangle$

Prediction

$$\left\langle \boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right), \boldsymbol{\phi}(\boldsymbol{x})\right\rangle$$

$$\hat{y} = \sum_{i=1}^{n} \alpha_i y^{(i)} \langle \boldsymbol{x}^{(i)}, \boldsymbol{x} \rangle + b$$

prediction depends only on inner products between test example and SVs $\langle \boldsymbol{x}^{(i)}, \boldsymbol{x} \rangle$

Kernels

- Capture nonlinear patterns in data
  - because linear models (e.g. regression, SVM) may not be rich enough
- **kernels** make linear models work in non-linear settings
  - by mapping to higher dimensions $x \longrightarrow \phi(x)$
  - and applying linear model in new input space
- Problem
  - computing mapping may be inefficient
  - using mapped features could be inefficient
- Solution: kernels!
  - mapping does not have to be explicitly computed
  - computations with mapped features remain efficient

The Quadratic Kernel

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \boldsymbol{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

(This slide intentionally left blank.)

Formal Definition

$$\phi : \mathcal{X} \mapsto \mathcal{F}$$
$$k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$$

ϕ takes input $\boldsymbol{x} \in \mathcal{X}$ (input space)
and maps to $\mathcal{F}$ (feature space)

kernel k takes two inputs $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{z} \in \mathcal{X}$
and computes their similarity $\langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{z}) \rangle$ in $\mathcal{F}$

Can any function be used as a kernel function?

Note: $\mathcal{F}$ needs to be a vector space with a dot product defined on it (aka a Hilbert space).

Valid Kernels

Kernel Matrix (aka **Gram Matrix**)

Kernel k also defines kernel matrix $\boldsymbol{K}$ over data

Given m examples (m finite, not necessarily training set),
let square $m \times m$ matrix $\boldsymbol{K}$ be defined so that

$$K_{ij} = k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \langle \boldsymbol{\phi}(\boldsymbol{x}^{(i)}), \boldsymbol{\phi}(\boldsymbol{x}^{(j)}) \rangle$$

Notes

- $\boldsymbol{K}$ is a $m \times m$ matrix of pairwise similarities
- $K_{ij} = K_{ji}$ (by symmetry of dot products) so $\boldsymbol{K}$ is symmetric

Theorem (Mercer)

Let $k : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ be given. Then for k to be a valid kernel, it is necessary and sufficient that for any $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\}$ (n finite), the corresponding kernel matrix $\boldsymbol{K}$ is symmetric positive semi-definite. [prove in homework]

Reminder: A matrix $\boldsymbol{K}$ is PSD if for all real vectors $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha} \geq 0$.

Kernels

Learning Goals

- Describe common kernels
  - linear, polynomial, Gaussian (RBF)
- Prove that RBF kernel is a valid kernel
  - using kernel closure properties

Polynomial Kernel

Let $k(\boldsymbol{x},\boldsymbol{z}) = (1 + \langle \boldsymbol{x},\boldsymbol{z} \rangle)^p$ (hyperparameter $p = 1,\ 2,\ \ldots$).
Then $\phi(\boldsymbol{x})$ contains all terms up to degree p.

Q: For $k(\boldsymbol{x},\boldsymbol{z}) = (1 + \langle \boldsymbol{x},\boldsymbol{z} \rangle)^2$, where $\boldsymbol{x},\boldsymbol{z} \in \mathbb{R}^2$,
what is the corresponding mapping $\phi(\boldsymbol{x})$?

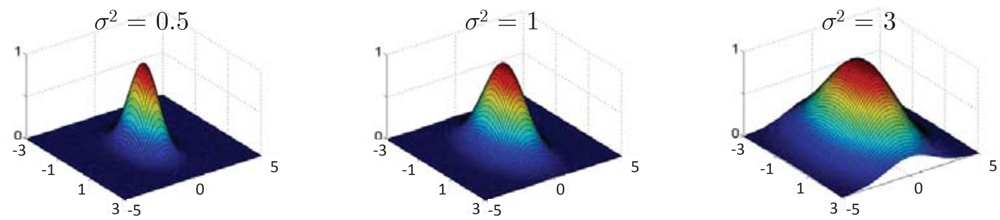
S:

scikit-learn: $k(\boldsymbol{x},\boldsymbol{z}) = (\gamma\langle \boldsymbol{x},\boldsymbol{z} \rangle + r)^d$
γ and r trade-off influence of lower-order terms

Gaussian Kernel
(aka Radial Basis Function Kernel)

$k(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|_2^2}{2\sigma^2}\right)$ (hyperparameter $\sigma^2 > 0$)

- Has value 1 when $\boldsymbol{x} = \boldsymbol{z}$, value falls off to 0 with increasing distance
- Interpreting σ^2



less "smooth" decision boundary
σ^2 too small $\Rightarrow$ overfit

smoother decision boundary
σ^2 too large $\Rightarrow$ underfit

Note: need to do feature scaling before using Gaussian Kernel

Is this a valid kernel?

scikit-learn: $k(\boldsymbol{x},\boldsymbol{z}) = \exp(-\gamma||\boldsymbol{x} - \boldsymbol{z}||^2)$, where $\gamma = 1/(2\sigma^2) > 0$

Images from Eric Eaton

Popular Kernels

- linear $\quad k(\boldsymbol{x}, \boldsymbol{z}) = \langle \boldsymbol{x}, \boldsymbol{z} \rangle$
- polynomial $\quad k(\boldsymbol{x}, \boldsymbol{z}) = (\gamma\langle \boldsymbol{x}, \boldsymbol{z} \rangle + r)^d$
- Gaussian (RBF) $\quad k(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|_2^2}{2\sigma^2}\right)$
- sigmoid $\quad k(\boldsymbol{x}, \boldsymbol{z}) = \tanh(\gamma\langle \boldsymbol{x}, \boldsymbol{z} \rangle + r)$
  - SVM with sigmoid kernel equivalent to 2-layer perceptron (neural network)
- cosine $\quad k(\boldsymbol{x}, \boldsymbol{z}) = \frac{\langle \boldsymbol{x}, \boldsymbol{z} \rangle}{\|\boldsymbol{x}\|\|\boldsymbol{z}\|} = \langle \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}, \frac{\boldsymbol{z}}{\|\boldsymbol{z}\|} \rangle$
  - popular choice for measuring similarity of text documents
  - normalizing (dividing by L_2-norm) projects vectors onto unit sphere, their dot product is the cosine of the angle between the vectors
- many more …

Based on slide by Eric Eaton

Kernel Construction

Instead of determining whether $k(\boldsymbol{x}, \boldsymbol{z})$ is a valid kernel, construct $k(\boldsymbol{x}, \boldsymbol{z})$ from simpler kernels (active area of ML).

Closure Properties of Kernels

Let $k_1(\boldsymbol{x}, \boldsymbol{z})$ and $k_2(\boldsymbol{x}, \boldsymbol{z})$ be valid kernels with feature mappings $\boldsymbol{\phi}^{(1)}(\boldsymbol{x})$ and $\boldsymbol{\phi}^{(2)}(\boldsymbol{x})$. Then

- (addition) $k(\boldsymbol{x}, \boldsymbol{z}) = k_1(\boldsymbol{x}, \boldsymbol{z}) + k_2(\boldsymbol{x}, \boldsymbol{z})$
- (scaling) $k(\boldsymbol{x}, \boldsymbol{z}) = f(\boldsymbol{x})\ k_1(\boldsymbol{x}, \boldsymbol{z})\ f(\boldsymbol{z})$
  for any real-valued function $f : \mathbb{R}^d \mapsto \mathbb{R}$
- (multiplication) $k(\boldsymbol{x}, \boldsymbol{z}) = k_1(\boldsymbol{x}, \boldsymbol{z})\ k_2(\boldsymbol{x}, \boldsymbol{z})$

are all valid kernels.

Proofs

Two options:

(1) Determine implicit feature mapping $\boldsymbol{\phi}(\boldsymbol{x})$.

(2) Show that kernel matrix is symmetric PSD.

Addition: $k(\boldsymbol{x}, \boldsymbol{z}) = k_1(\boldsymbol{x}, \boldsymbol{z}) + k_2(\boldsymbol{x}, \boldsymbol{z})$

Prove that RBF Kernel is a Valid Kernel

Assume $\sigma^2 = 1$.

(Easy to generalize, or recognize $ak(\boldsymbol{x}, \boldsymbol{z})$ for $a > 0$ is a valid kernel.)

Based on notes by Tommi Jaakola

SVMs

Learning Goals

- ✓ Solving the SVM optimization problem
  - using Lagrange multipliers (leads to dual problem)
- ✓ Allowing misclassified examples
  - using slack variables (leads to soft-margin SVM)
- ✓ Describe the SVM loss function
- ✓ Allowing non-linear decision boundaries
  - using kernels

Take-Aways

- Maximum-margin separator
- Primal-dual formulation
- Hard vs soft-margin SVM
- Hinge loss
- Kernels (“kernel trick”)

SVM Practical Advice

When Applying SVMs

Use SVM software package to solve for parameters

- e.g. SVMlight, libsvm, cvx (fast!), etc

Need to specify

- Choice of hyperparameter C
- Choice of kernel function
- Associated kernel parameters
  e.g. p for polynomial kernel, σ for RBF kernel

Based on slide by Eric Eaton

Practical Advice

When faced with ML problem, it is sometimes not clear which algorithm to use.

The algorithm matters, but what matters more are

- How much data you have
- How good you are at error analysis and debugging learning algorithms
- How you design features
- (Upcoming lecture: Advice for Applying ML)