Recitation 5

Kernels

DS-GA 1003 Machine Learning

CDS

February 22, 2023

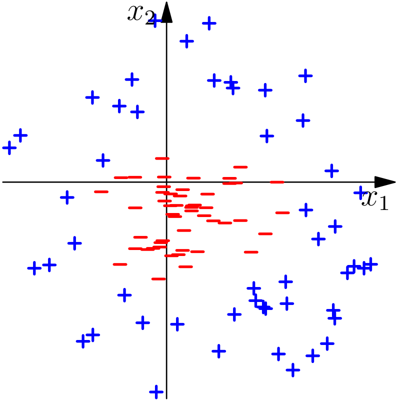
Outline

- Motivation
- Kernel
- Incorporating kernels into ridge regression and SVM
- RBF Kernel
- IPython Demo: MNIST with RBF kernel

Motivation

Question

Consider applying SVM to the data set. What is the issue?



Motivation

Question

Consider applying SVM to the data set. What is the issue?

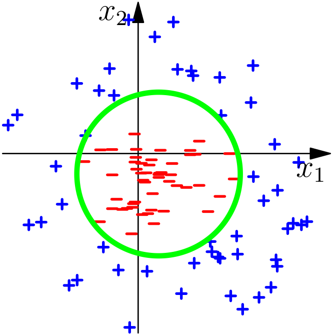
Solution

We want to allow for non-linear regression functions, but we would like to reuse the same fitting procedures we have already developed. To do this we will expand our feature set by adding non-linear functions of old features.

Motivation

Solution

For the SVM we expand our feature vector from $(1, x_1, x_2)$ to $(1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$. Using $w = (-1.875, 2.5, -2.5, 0, 1, 1)$ gives $-1.875 + 2.5x_1 - 2.5x_2 + x_1^2 + x_2^2 = (x_1 + 1.25)^2 + (x_2 - 1.25)^2 - 5 = 0$ as our decision boundary.



Motivation

- Linear model is clearly insufficient to represent these problems.
- The most intuitive solution is to **expand the input space**
  - Adding features
- We can define a **feature map function** $\varphi(x) : \mathcal{X} \mapsto \mathcal{H}$
  - $dim(\mathcal{H}) > dim(\mathcal{X})$
  - For SVM example above, $\varphi(1, x_1, x_2) = [1, x_1, x_2, x_1x_2, x_1^2, x_2^2]$.
- We then find a linear separator on the feature space $\mathcal{H}$.

Adding Features

- Polynomials can approximate any function (Taylor's Theorem).
- We can linearly model any problem perfectly if we add enough terms.
- But adding features obviously comes with a cost.
- The cost grows exponentially as we increase the degree.

Adding Features

Question

Suppose we begin with d-dimensional inputs $x = (x_1, \ldots, x_d)$. We add all features up to degree M. More precisely, all terms of the form

$$x_1^{p_1} \cdots x_d^{p_d} \quad p_i \geq 0 \text{ and } p_1 + \cdots + p_d \leq M$$

How many features will we have in total?

- There will be $\binom{M+d}{M}$ terms total. Grows very quickly!
  - For example, if $d = 40$ and $M = 8$ we get $\binom{40+8}{8} = 377348994$.
- Both M and d impacts the cost of adding features.
- If we stick with polynomial features up to order M, it's takes exponential time $O(d^M)$ to compute all features.
- **How do we make the computation feasible?**

Representer Theorem

Theorem

Suppose you have a loss function of the form

$$J(w) = L(w^T\varphi(x_1), \ldots, w^T\varphi(x_n)) + R(\|w\|_2)$$

where

- $x_i \in \mathbb{R}^d, w \in \mathbb{R}^{d'}, \varphi(x) : \mathbb{R}^d \mapsto \mathbb{R}^{d'}$.
- $L : \mathbb{R}^n \to \mathbb{R}$ *is an arbitrary function (loss term).*
- $R : \mathbb{R}_{\geq 0} \to \mathbb{R}$ *is increasing (regularization term).*

Assume J has at least one minimizer. Then J has a minimizer w^* of the form $w^* = \sum_{i=1}^n \alpha_i \varphi(x_i)$ for some $\alpha \in \mathbb{R}^n$. If R is strictly increasing, then all minimizers have this form.

Representer Theorem: Proof

Proof.

- Let $w^* \in \mathbb{R}^{d'}$ and let $S = \text{Span}(\varphi(x_1), \ldots, \varphi(x_n))$.
- Suppose w^* is the optimal parameter, and it **does not lie in** S.
- Then we can write $w^* = u + v$ where $u \in S$ and $v \in S^\perp$. (Here u is the orthogonal projection of w^* onto S, and $S^\perp$ is the subspace of all vectors orthogonal to S.)
- Then $(w^*)^T\varphi(x_i) = (u+v)^T\varphi(x_i) = u^T\varphi(x_i) + v^T\varphi(x_i) = u^T\varphi(x_i)$. So the prediction only depends on $u^T\varphi(x_i)$.
- But $\|w^*\|_2^2 = \|u+v\|_2^2 = \|u\|_2^2 + \|v\|_2^2 + 2u^Tv = \|u\|_2^2 + \|v\|_2^2 \geq \|u\|_2^2$.
- Thus $R(\|w^*\|_2) \geq R(\|u\|_2)$ showing $J(w^*) \geq J(u)$.

□

Representer Theorem

- If your loss function only depends on w via its inner products with the inputs, and the regularization is an increasing function of the ℓ_2 norm, then we can write w^* as a linear combination of the training data.
- This applies to ridge regression and SVM.

Representer Theorem: Ridge Regression

- By adding features to ridge regression we had

$$\begin{aligned} J(\tilde{w}) &= \frac{1}{n}\sum_{i=1}^{n}(\tilde{w}^T\varphi(x_i)-y_i)^2+\lambda\|\tilde{w}\|_2^2 \\ &= \frac{1}{n}\|\Phi\tilde{w}-y\|_2^2+\lambda\tilde{w}^T\tilde{w}, \end{aligned}$$

  where $\Phi\in\mathbb{R}^{n\times d'}$ is the matrix with $\varphi(x_i)^T$ as its ith row.
- Representer Theorem applies giving $\tilde{w}=\sum_{j=1}^{n}\alpha_j\varphi(x_j)=\Phi^T\alpha$.
- Plugging in gives

$$J(\alpha)=\frac{1}{n}\left\|\Phi\Phi^T\alpha-y\right\|_2^2+\lambda\alpha^T\Phi\Phi^T\alpha.$$

- Define $K=\Phi\Phi^T$

Representer Theorem: Dual SVM

- The dual SVM problem (with features) is given by

$$\begin{aligned}
\text{maximize}_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \varphi(x_i)^T \varphi(x_j) \\
\text{subject to} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\
& \alpha_i \in \left[0, \frac{c}{n}\right] \quad \text{for } i = 1, \ldots, n.
\end{aligned}$$

- We can immediately kernelize (no representer theorem needed) by replacing $\varphi(x_i)^T \varphi(x_j) = k(x_i, x_j)$.
- Recall that we were able to derive the conclusion of the representer theorem using strong duality for SVMs.

The Kernel Function

Definition (Kernel)

Given a feautre map $\varphi(x) : \mathcal{X} \mapsto \mathcal{Z}$, the **kernel function** corresponding to $\varphi(x)$ is

$$k(x, x^{'}) = \langle \varphi(x), \varphi(x^{'}) \rangle$$

where $\langle \cdot, \cdot \rangle$ is an inner product operator.

- So a kernel function computes the inner product of applying the feature map $\varphi(x)$ for two inputs $x, x' \in \mathcal{X}$.
- We only need to know the output of the kernel to find the parameters.
- Predictor function is:

$$f(x^*) = \sum_i \alpha_i k(x_i, x^*)$$

Efficiency of Kernel

Consider the polynomial kernel $k(x, y) = \langle \varphi(x), \varphi(y) \rangle = (1 + x^T y)^M$ where $x, y \in \mathbb{R}^d$. For example, if $M = 2$ we have

$$\begin{aligned} (1 + x^T y)^2 &= 1 + 2x^T y + x^T y x^T y \\ &= 1 + 2 \textstyle\sum_{i=1}^d x_i y_i + \sum_{i,j=1}^d x_i y_i x_j y_j . \end{aligned}$$

Option 1: First explicitly evaluate $\varphi(x)$ and $\varphi(y)$, and then compute $\langle \varphi(x), \varphi(y) \rangle$.

- $\varphi(x) = (1, \sqrt{2}x_1, \ldots, \sqrt{2}x_d, x_1^2, \ldots, x_d^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \ldots, \sqrt{2}x_{d-1}x_d)$
- Takes $O(d^M)$ times to evaluate $\varphi(x)$ and $\varphi(y)$.
- Takes another $O(d^M)$ times to compute the inner product.
- Time complexity is $O(d^M)$.

Efficiency of Kernel

Consider the polynomial kernel $k(x, y) = \langle \varphi(x), \varphi(y) \rangle = (1 + x^T y)^M$ where $x, y \in \mathbb{R}^d$. This computes the inner product of all monomials up to degree M in time $O(d)$. For example, if $M = 2$ we have

$$\begin{aligned}(1 + x^T y)^2 &= 1 + 2x^T y + x^T y x^T y \\ &= 1 + 2\sum_{i=1}^d x_i y_i + \sum_{i,j=1}^d x_i y_i x_j y_j.\end{aligned}$$

Option 2: First calculate $1 + x^T y$, then calculate $(1 + x^T y)^M$.

- Takes $O(d)$ time to evaluate $1 + x^T y$.
- Takes $O(1)$ time to calculate $(1 + x^T y)^M$
- Time complexity is $O(d)$

Recap on what we achieved

- Start with a low dimensional model
    - Due to limited input data size
    - Number of parameters is d
- Want to increase the model capacity by adding features $x_i \to \varphi(x_i)$
    - The cost is too high as we increase degrees
    - Number of parameters is $d', d' >> d$
- Realize the optimal parameter is a linear combination of $\varphi(x_i)$
    - Representer Theorem
    - Number of parameters becomes $N, d' >> N > d$
- Realize we only need the inner product of two $\varphi(x_i)$, $k(\cdot, \cdot)$
    - There are more efficient methods to compute the inner product
    - We don't need to explicitly compute $\varphi(\cdot)$
- The rephrased problem becomes a linear problem
    - But the solution still has high dimensional expressive power!

Mercer's Theorem

- Not all function $f(x, y)$ are valid kernels.
- How can we know if $k(x, y)$ is a valid kernel or not?

Theorem (Mercer's Theorem)

Fix a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. There is a Hilbert space H and a feature map $\varphi : \mathcal{X} \to H$ such that $k(x, y) = \langle \varphi(x), \varphi(y) \rangle_H$ if and only if for any $x_1, \ldots, x_n \in \mathcal{X}$ the associated matrix K is positive semi-definite:

$$K = \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{pmatrix}.$$

Such a kernel k is called **positive semi-definite***.*

Positive Semi-Definite

Definition (Positive Semi-Definite)

A matrix $A \in \mathbb{R}^{n \times n}$ is **positive semi-definite** if it is symmetric and

$$x^T A x \geq 0$$

for all $x \in \mathbb{R}^n$.

- Equivalent to saying the matrix is symmetric with non-negative eigenvalues.

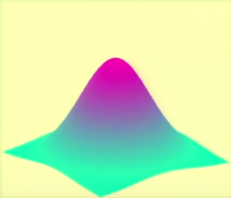
Kernel Examples

- Dot Product
  - $k(x_i, x_j) = x_i^T x_j$
- Mth Polynomial Kernels
  - $k(x_i, x_j) = (1 + x_i^T x_j)^M$
- RBF Kernels
  - $k(x_i, x_j) = exp(-\frac{||x_i - x_j||^2}{2\sigma^2})$
- Sigmoid kernel
  - $k(x_i, x_j) = tanh(\alpha x_i^T x_j + c)$

RBF Kernels

$$k(w, x) = \exp\left(-\frac{\|w - x\|_2^2}{2\sigma^2}\right).$$

- 2d RBF kernel looks like the following.
- Let's say we fix w. The $k(w, x)$ is high when x is very close to w. The value decays as x is moving away from w.
- σ controls the spread of the kernel. The higher σ is the wider / flatter the landscape is for $k(w, x)$.



RBF Kernels

- As we saw earlier for ridge regression and SVM classification, the decision function has the form $f_\alpha(x) = \sum_{i=1}^n \alpha_i k(x_i, x)$.
- For ridge regression, this means that using the RBF kernel amounts to approximating our data by a linear combination of Gaussian bumps.
- For SVM classification, each $k(x_i, x) = \exp\left(-\|x_i - x\|_2^2/(2\sigma^2)\right)$ represents a exponentially decaying distance between x_i and x. Thus our decisions depend on our proximities to data points.

Going to infinite dimension

- What is the polynomial expression of $\varphi(\cdot)$ for RBF and Sigmoid Kernel?
  - There are no finite expression, they are sum of infinite polynomials
  - $\varphi(x) = e^{-x^2/2\sigma^2} \left[1, \sqrt{\frac{1}{1!\sigma^2}}x, \sqrt{\frac{1}{2!\sigma^4}}x^2, \sqrt{\frac{1}{3!\sigma^6}}x^3, \ldots\right]$
- This implies we have essentially modeled the problem using a infinite degree polynomial!
- At this point, the factor limiting our model capacity is the amount of training data.

Finding Your Own Kernels

Let $k_1, k_2 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be positive semi-definite kernels. Then so are the following:

- $k_3(w, x) = k_1(w, x) + k_2(w, x)$
- $k_4(w, x) = \alpha k_1(w, x)$ for $\alpha \geq 0$
- $k_5(w, x) = f(w)f(x)$ for any function $f : \mathcal{X} \to \mathbb{R}$
- $k_6(w, x) = k_1(w, x)k_2(w, x)$

Remarks

- With Representer Theorem, we can re-parameterize our prediction function from $f_w(x) = w^T\varphi(x)$ to $f_\alpha(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x)$.
- The feature representation $\varphi(x)$ only appears in inner product form in both the loss function and the prediction function.
- Therefore, we just need to evaluate the kernel function $k(x, y)$ and never need to explicitly evaluate $\varphi(x)$. It's much easier to compute the kernel $k(x, y)$ than the inner product.
- The kernel $k(x, y)$, to some extent, represents a similarity score between two data points.
- We are almost guaranteed to overfit on training data, so regularization is very important.

References

- DS-GA 1003 Machine Learning Spring 2021