Discussion

He He

CDS, NYU

March 2, 2021

Motivation for kernels

- Our data is typically not linearly separable.
- But we like to work with linear models.
- Adding features (going to high-dimensional space) allow us to use linear models for complex data.
- Kernels allow us to think about similarities rather than feature engineering.

1. For NLP problems, kernels are less popular because we already have large feature vector by introducing feature templates. It's also computationally less costly since the feature vectors are sparse, compared to kernel methods (e.g. tree kernel).

Two perspectives on kernels

- Given a feature map $\phi : \mathcal{X} \to \mathcal{H}$, we can define a kernel function $k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$.
- Given a PD kernel $k : \mathcal{X} \times \mathcal{X} \to R$, there exists a corresponding feature map.
  - Note that the kernel does not uniquely define the feature map.
- In practice we typically only work with the kernel function.

RBF Kernel

RBF Basis

Input space $\mathcal{X} = \mathrm{R}^d$

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

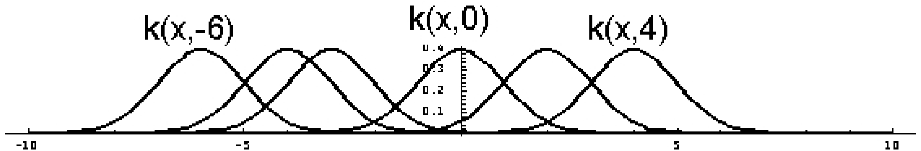
where σ^2 is known as the bandwidth parameter.

- Suppose we have 6 training examples: $x_i \in \{-6, -4, -3, 0, 2, 4\}$.
- If representer theorem applies, then

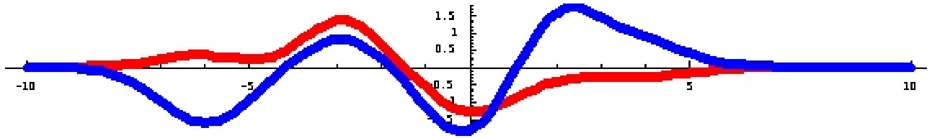
$$f(x) = \sum_{i=1}^{6} \alpha_i k(x_i, x).$$

RBF Predictions

- f is a linear combination of 6 basis functions of form $k(x_i, \cdot)$:



- Predictions of the form $f(x) = \sum_{i=1}^{6} \alpha_i k(x_i, x)$:



- When kernelizing with RBF kernel, prediction functions always look this way (whether we get w from SVM, ridge regression).

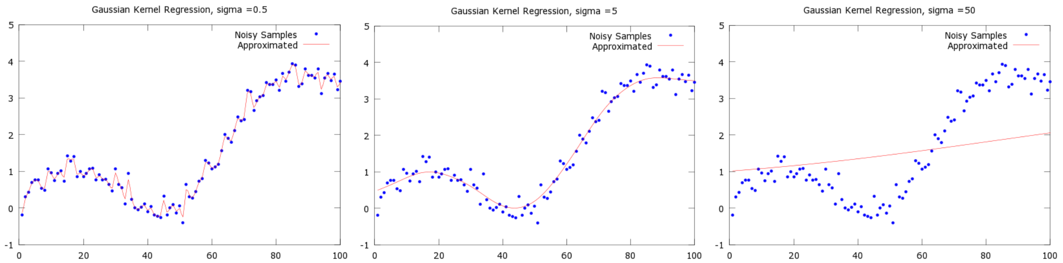
Effect of the bandwidth

How does the fitted function change when we vary the bandwidth parameter?

https://mccormickml.com/2014/02/26/kernel-regression/

Effect of the bandwidth

How does the fitted function change when we vary the bandwidth parameter?



(a) $\alpha = 0.5$ (b) $\alpha = 5$ (c) $\alpha = 50$

https://mccormickml.com/2014/02/26/kernel-regression/

Feature map of RBF kernel

What feature map corresponds to the RBF kernel?

Consider the 1D case ($x \in \mathbb{R}$) where $\sigma = 1$:

$$k(x,x') = \exp\left(-\frac{(x-x')^2}{2}\right) \tag{1}$$

$$= \exp\left(-\frac{x^2}{2}\right)\exp\left(xx'\right)\exp\left(-\frac{x'^2}{2}\right) \tag{2}$$

Based on https://www.cs.ubc.ca/~schmidtm/Courses/540-W19/L12.5.pdf

1. Given

$$k(x,x') = \exp\left(-\frac{(x-x')^2}{2}\right) \tag{3}$$

$$= \exp\left(-\frac{x^2}{2}\right)\exp\left(xx'\right)\exp\left(-\frac{x'^2}{2}\right) \tag{4}$$

2. We can also prove here that the Gaussian kernel is PD. The first and the third term is a 1D feature map, so their product is a PD kernel. The middle term is $\exp^{\langle x,x'\rangle}$, which is also a PD kernel. We can prove $\exp^{k(x,x')}$ is a kernel by writing e as a power series, then each term in the sum is a product of valid kernels.
3. Now we only need to make the middle term separable in x and x', i.e. $\exp(xx') = g(x)g(x')$.
4. Recall that

$$e^x := \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

.

5. So we have $g(x)_k = \frac{x^k}{\sqrt{k!}}$ and $e^{xx'} = \sum_{k=0}^{\infty} g(x)_k g(x')_k$.
6.

$$\phi(x)_k = \exp\left(-\frac{x^2}{2}\right)\frac{x^k}{\sqrt{k!}}$$

Kernel Methods

Kernelization

- A method can be kernelized if both training and inference only need inner produc in the feature space.
- Representer theorem says that all norm-regularized linear models can be kernelized.
  - Although we might be in a high dimensional space, w lies in the subspace spanned by $\phi(x_i)$.
  - Dimension of the subspace grows with the dataset size.
- Many other algorithms can be kernelized.

1. Note that the norm is induced by the Hilbert space.

Kernelized perceptron

- Initialize $w \leftarrow 0$
- While not converged
    - For $(x_i, y_i) \in \mathcal{D}$
        - If $y_i w^T x_i < 0$
        - Update $w \leftarrow w + y_i x_i$

1. Show that w is in the span of data by induction: each update leaves w in the span of data.
2. Prediction:
$$w \cdot x = \sum_{i=1}^{n} \alpha_i x_i \cdot x = k_x^T \alpha$$
3. Training: $\alpha_i \leftarrow \alpha_i + y_i$

Other kernel methods

- Distance-based methods depending on $\|x - x'\|^2$
  - k-means clustering
  - k-nearest neighbors
- Eigenvalue methods: can show that eigenvector is in the span of data
  - Principal component analysis
  - Spectral clustering

1.
$$\|x - x'\|^2 = \langle x - x', x - x' \rangle = \langle x, x \rangle - 2\langle x, x' \rangle + \langle x', x' \rangle$$

Kernel SVM vs ridge

- For both kernel SVM and ridge regression, we make predictions by

$$\hat{f}(x) = k_x^T \alpha^* = \sum_{i=1}^{n} \alpha_i^* k(x_i, x)$$

- For SVM, we have sparsity in α^* from complementary slackness.
- For ridge, we need to access all training examples.
- For large-scale dataset, we may not be able to store/compute the kernel matrix.
    - Large-scale kernel machines (e.g. Random Features for Large-Scale Kernel Machines)