Advanced Machine Learning Subsidary Notes

Lecture 18: Kernel Trick

Adam Prügel-Bennett

March 8, 2024

1 Keywords

- The Kernel Trick, SVMs, Regression

2 Main Points

2.1 Kernels

- SVM kernels are symmetric functions of two variable that can be factorised as an inner-product

$$K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\phi}(\boldsymbol{x})^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{y}) = \sum_{i=1}^{p'} \phi_i(\boldsymbol{x})\, \phi_i(\boldsymbol{y})$$

  - $\boldsymbol{\phi}(\boldsymbol{x})$ are vectors whose elements, $\phi_i(\boldsymbol{x})$, are real-valued functions of the features $\boldsymbol{x}$ (every different feature will correspond to a different vector $\boldsymbol{\phi}(\boldsymbol{x})$)
  - p' is the dimensionality of the extended feature space which might be infinite
  - An immediate consequence of this is that the vectors are positive semi-definite
    * This follows because for any function $f(\boldsymbol{x})$ the quadratic form is non-negative

$$\begin{aligned}\iint f(\boldsymbol{x})\, K(\boldsymbol{x}, \boldsymbol{y})\, f(\boldsymbol{y}) \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y} &= \iint f(\boldsymbol{x})\, \boldsymbol{\phi}(\boldsymbol{x})^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{y})\, f(\boldsymbol{y})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y} \\ &= \sum_{i=1}^{p'} \left(\int f(\boldsymbol{x})\, \phi_i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \right)^2 \geq 0\end{aligned}$$

- **Eigenfunctions and Mercer's Theorem**
  - Kernel functions play the same role for functions as matrices do for normal vectors
    * that is they describe general linear transformations
    * for a function $f(\boldsymbol{x})$ the argument $\boldsymbol{x}$ can be seen as an index just like i is the index of element v_i of a vector $\boldsymbol{v}$
    * we will consider only symmetric kernels (that is, where $K(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{y}, \boldsymbol{x})$
    * these play a similar role as symmetric matrices
  - Eigensystems for Kernels
    * $\psi(\boldsymbol{y})$ is said to be an eigenfunction of a kernel functions if

$$\int K(\boldsymbol{x}, \boldsymbol{y})\, \psi(\boldsymbol{y})\, \mathrm{d}\, \boldsymbol{y} = \lambda\, \psi(\boldsymbol{x})$$

    * In an analogy to the eigen-decomposition of a symmetric matrix we can define the eigen-decomposition of a symmetric kernel function

$$K(\boldsymbol{x}, \boldsymbol{y}) = \sum_i \lambda_i\, \psi_i(\boldsymbol{x})\, \psi_i(\boldsymbol{y})$$

* This is known as **Mercer's Theorem**
* We proved the decomposition for matrices
  · the difficult part in the proof is that you need the eigenvectors to span the vector space
  · this is intuitively obvious if there are n orthogonal eigenvectors in an n-dimensional space
  · it is harder in functions spaces and you need to define the vector space you are modelling (e.g. L_2)
  · if you assume that the set of eigenvectors span the function space then the rest of the proof is the same as for matrices
  · don't worry if you don't understand this it is enough to remember Mercer's Theorem
* Mercer's Theorem holds for any symmetric kernel function (it does not have to be positive semi-definite)
* But if $K(\boldsymbol{x}, \boldsymbol{y})$ are positive semi-define then there exist real functions $\phi_i(\boldsymbol{x}) = \sqrt{\lambda_i}\psi_i(\boldsymbol{x})$ such that

$$K(\boldsymbol{x}, \boldsymbol{y}) = \sum_i \lambda_i \, \psi_i(\boldsymbol{x}) \, \psi_i(\boldsymbol{y}) = \sum_i \phi_i(\boldsymbol{x}) \, \phi_i(\boldsymbol{y})$$

  · if $K(\boldsymbol{x}, \boldsymbol{y})$ was not positive semi-definite then some of the eigenvalues would be negative and the functions $\psi_i(\boldsymbol{x})$ would not be real-valued

2.2 SVM Kernels

- SVM Kernels are positive semi-definite symmetric functions
  - There are four necessary and sufficient conditions that hold for any positive semi-definite kernel
    1. All their eigenvalues are non-negative (i.e. either zero or positive)
    2. They can be decomposed as

$$K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\phi}(\boldsymbol{x})^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{y}) = \sum_{i=1}^{p'} \phi_i(\boldsymbol{x}) \, \phi_i(\boldsymbol{y})$$

       where $\phi_i(\boldsymbol{x})$ are real-valued functions
    3. Their quadratic form with any function $f(\boldsymbol{x})$ is non-negative
    4. For any set of points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ the matrix

$$\mathbf{K} = \begin{pmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}_1) & K(\boldsymbol{x}_1, \boldsymbol{x}_2) & \cdots & K(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ K(\boldsymbol{x}_2, \boldsymbol{x}_1) & K(\boldsymbol{x}_2, \boldsymbol{x}_2) & \cdots & K(\boldsymbol{x}_2, \boldsymbol{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_n, \boldsymbol{x}_1) & K(\boldsymbol{x}_n, \boldsymbol{x}_2) & \cdots & K(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{pmatrix}$$

       is a positive semi-definite matrix
       * such matrices are known as **Gram matrices**
       * I didn't mention this in the lecture and won't use this property, but for completeness I mention it here (you won't be tested on it)
       * the proof that this is a necessary condition follows rather simply from the fact that if we define a matrix $\mathbf{\Phi}$ with elements $\Phi_{ik} = \phi_i(\boldsymbol{x}_k)$ then $\mathbf{K} = \mathbf{\Phi}^{\mathsf{T}}\mathbf{\Phi}$ and we have seen many times any such matrix is positive semi-definite
- Recall from the previous lecture that any kernel function that allows a decomposition in terms of positive functions can be used an SVM where we can use the kernel trick
  - If we don't use positive semi-definite kernels then our "distances" (used in computing margins) are no-longer proper distances and can be negative (invalidating everything)

2.3 Constructing SVM Kernel

- Most functions of two variable won't be positive semi-definite
- Given a function of two variables it is hard to determine if it is positive-semi definite (none of the definitions are particularly easy to use)
- However we can use simple rules to build positive-semi definite (PSD) kernels from other positive semi-definite kernels

1. Our starting point is to note the inner produce $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^\mathsf{T} \boldsymbol{y}$ is positive semi-definite
   - as an aside we don't necessarily need to use normal vectors as our features so long as we objects with an inner-product
2. Adding PSD kernels
   if $K_1(\boldsymbol{x}, \boldsymbol{y})$ and $K_2(\boldsymbol{x}, \boldsymbol{y})$ are PSD kernels then so is $K_3(\boldsymbol{x}, \boldsymbol{y}) = K_1(\boldsymbol{x}, \boldsymbol{y}) + K_2(\boldsymbol{x}, \boldsymbol{y})$
   - To prove this we can use the property that PSD have non-negative quadratic form

$$
\begin{aligned}
Q &= \int f(\boldsymbol{x}) \, K_3(\boldsymbol{x}, \boldsymbol{y}) \, f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\boldsymbol{y} \\
&= \int f(\boldsymbol{x}) \, \big(K_1(\boldsymbol{x}, \boldsymbol{y}) + K_2(\boldsymbol{x}, \boldsymbol{y})\big) \, f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\boldsymbol{y} \\
&= \int f(\boldsymbol{x}) \, K_1(\boldsymbol{x}, \boldsymbol{y}) \, f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\boldsymbol{y} + \int f(\boldsymbol{x}) \, K_2(\boldsymbol{x}, \boldsymbol{y}) \, f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\boldsymbol{y} \geq 0
\end{aligned}
$$

3. Multiplication by a positive scalar
   if $K_1(\boldsymbol{x}, \boldsymbol{y})$ is a PSD kernels and $c > 0$ then so is $K_3(\boldsymbol{x}, \boldsymbol{y}) = c \, K_1(\boldsymbol{x}, , \boldsymbol{y})$
   - We can prove this in a similar way to the last proof
4. Multiply PSD kernels
   if $K_1(\boldsymbol{x}, \boldsymbol{y})$ and $K_2(\boldsymbol{x}, \boldsymbol{y})$ are PSD kernels then so is $K_3(\boldsymbol{x}, \boldsymbol{y}) = K_1(\boldsymbol{x}, \boldsymbol{y}) \, K_2(\boldsymbol{x}, \boldsymbol{y})$
   - This is easy to prove using the decomposition of PSD to inner products

$$
K_3(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i,j} \phi_i^1(\boldsymbol{x}) \, \phi_i^1(\boldsymbol{y}) \, \phi_j^2(\boldsymbol{x}) \, \phi_j^2(\boldsymbol{y}) = \sum_{i,j} \phi_{ij}^3(\boldsymbol{x}) \, \phi_{ij}^3(\boldsymbol{y})
$$

   where $\phi_{ij}^3(\boldsymbol{x}) = \phi_i^1(\boldsymbol{x}) \, \phi_j^2(\boldsymbol{x})$
     * this (double) sum we can treat as an inner-product
     * if is easy to show that the quadratic form with any function $f(\boldsymbol{x})$ is non-negative
5. Powers of PSD kernels
   if $K_1(\boldsymbol{x}, \boldsymbol{y})$ is a PSD kernels then so is $K_1^n(\boldsymbol{x}, \boldsymbol{y})$ for any natural number n
   - Since the product of any two PSD kernels are PSD then the square of a PSD kernel is PSD
   - But by an inductive argument this holds for any integer power
6. Exponentiial of PSD kernels
   The exponential of a PSD kernel is also a PSD kernel
   - convergent Taylor expansions allow us to approximate a function to any degree of accuracy
   - often Taylor expansions aren't everywhere convergent (so we have to be careful
   - but Taylor expansions of exponentials are everywhere convergent
   - further Taylor expanding an exponential of a PSD kernel involves a sum of PSD kernels

$$
\mathrm{e}^{K(\boldsymbol{x}, \boldsymbol{y})} = \sum_i \frac{1}{i!} K^i(\boldsymbol{x}, \boldsymbol{y}) = 1 + K(\boldsymbol{x}, \boldsymbol{y}) + \frac{1}{2} K^2(\boldsymbol{x}, \boldsymbol{y}) + \cdots
$$

     * each term is a PSD kernel

7. If $K(\boldsymbol{x},\boldsymbol{y}) \succeq 0$ then $K_1(\boldsymbol{x},\boldsymbol{y}) = g(\boldsymbol{x})\,K(\boldsymbol{x},\boldsymbol{y})\,g(\boldsymbol{y}) \succeq 0$
   - Consider the quadratic form for any function $f(\boldsymbol{x})$

$$Q = \int f(\boldsymbol{x})\,K_1(\boldsymbol{x},\boldsymbol{y})\,f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} = \int f(\boldsymbol{x})\,g(\boldsymbol{x})\,K(\boldsymbol{x},\boldsymbol{y})\,g(\boldsymbol{y})\,f(\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y}$$
$$= \int h(\boldsymbol{x})\,K(\boldsymbol{x},\boldsymbol{y})\,h(\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} > 0$$

   where $h(\boldsymbol{x}) = f(\boldsymbol{x})\,g(\boldsymbol{x})$ and we used the fact that $K(\boldsymbol{x},\boldsymbol{y}) \succeq 0$ so that its quadratic form with any function (in this case $h(\boldsymbol{x})$) is non-negative.
   - As $Q \geq 0$ then $K_1(\boldsymbol{x},\boldsymbol{y}) \succeq 0$

- Using these properties we see that $K(\boldsymbol{x},\boldsymbol{y}) = \mathrm{e}^{-\gamma\,\|\boldsymbol{x}-\boldsymbol{y}\|^2}$ is a PSD kernel if $\gamma > 0$
  - Now $\boldsymbol{x}^\mathsf{T}\boldsymbol{y}$ is and inner product so a PSD kernel
  - Since $2\,\gamma > 0$ then $2\,\gamma\,\boldsymbol{x}^\mathsf{T}\boldsymbol{y}$ is a PSD kernel
  - But then so is $\mathrm{e}^{2\,\gamma\,\boldsymbol{x}^\mathsf{T}\boldsymbol{y}}$ as it is the exponential of a PSD kernel
  - Now
  $$K_1(\boldsymbol{x},\boldsymbol{y}) = \mathrm{e}^{-\gamma\,\|\boldsymbol{x}-\boldsymbol{y}\|^2} = \mathrm{e}^{-\gamma\,\|\boldsymbol{x}\|^2}\,\mathrm{e}^{2\,\gamma\,\boldsymbol{x}^\mathsf{T}\boldsymbol{y}}\,\mathrm{e}^{-\gamma\,\|\boldsymbol{y}\|^2}$$
  is of the form $g(\boldsymbol{x})\,K(\boldsymbol{x},\boldsymbol{y})\,g(\boldsymbol{y})$ where $K(\boldsymbol{x},\boldsymbol{y}) = \mathrm{e}^{2\,\gamma\,\boldsymbol{x}^\mathsf{T}\boldsymbol{y}} \succeq 0$ and $g(\boldsymbol{x}) = \mathrm{e}^{-\gamma\,\|\boldsymbol{x}\|^2}$ so $K_1(\boldsymbol{x},\boldsymbol{y}) \succeq 0$
  - This kernel is known as the *radial basis function* or *RBF* or *Gaussian kernel*
  - It has a hyper-parameter, γ that determines the length scale in the problem (or rather inverse-length scale)
  - this is a very important kernel as it often (but certainly not always) gives good performance (if γ is appropriately chosen)

- **Non-numerical Kernels**
  - When SVMs were fashionable there was a whole industry of researchers finding clever kernels
  - When working with language or trees or graphs it paid to create bespoke kernels for these structures
  - Typically these would all be built up from inner-products
  - Using clever algorithms you can build very clever kernels functions
  - One down side of SVM kernels is they don't naturally capture prior knowledge about the problem being tackled
    * a clever work around is to build SVMs based on other learning machines that are trained the problem
    * an example of this is the use of *Fisher kernels* based on Fisher information

2.4 Why SVM kernels need to be positive semi-definite

- Recall that we required $K(\boldsymbol{x},\boldsymbol{y}) = \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{y})\rangle$ where the components of $\boldsymbol{\phi}(\boldsymbol{x})$ are real, i.e. $\phi_k(x)$ is a real number.

- We used the inner product to compute the separation between the data points $\boldsymbol{\phi}(\boldsymbol{x}_i)$ and the separating plan given by $\boldsymbol{w}^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{x}) = b$ in the extended feature space.

- But this only makes sense if $\langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{y})\rangle$ defines a proper inner produce. In particular $\langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x})\rangle = \|\boldsymbol{\phi}(\boldsymbol{x})\|^2 \geq 0$ (with equality only if $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{0}$).

- Now by Mercer's theorem $\phi_i(\boldsymbol{x}) = \sqrt{\lambda_i}\,\psi_i(\boldsymbol{x})$ where $\psi_i(\boldsymbol{x})$ is the i^{th} eigenfunction of $K(\boldsymbol{x},\boldsymbol{y})$ with eigenvalue λ_i. But if $\lambda_i < 0$ then $\sqrt{\lambda_i}$ is imaginary. Note that

$$\|\boldsymbol{\phi}(\boldsymbol{x})\|^2 = \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x})\rangle = \sum_i \lambda_i\,\psi_i^2(\boldsymbol{x})$$

  but this is not guaranteed to be non-negative for every value of $\boldsymbol{x}$ if some eigenvalues are negative.

- You can also show that the Hessian of the dual objective function is equal to $-y_i\,y_j\,K(\boldsymbol{x}_i,\boldsymbol{x}_j)$ which is negative semi-definite provided $K(\boldsymbol{x},\boldsymbol{y}) \succeq 0$. If this is not the case there will be a maximum as some combination of the Lagrange multipliers that goes to infinity.

2.5 Beyond SVMs

- There are a lot of other kernel based learning machines
- Many of these use constraints
- They often involve linear operations between vectors where the optimum depends on the inner-product of vectors
  - thus we can use the kernel trick
- *SVR* are support vector machines for regression
  - here we try to find a dividing plane so that all points lie within a margin (the exact opposite of what we had)
  - We can introduce slack variables to allow some points to lie outside the margin
    - the slack variables much be non-negative
    - we can use a linear punishment s_i or quadratic punishment s_i^2
- We can also do *kernel ridge regression*

$$\min_{\boldsymbol{w}} \lambda\,\|\boldsymbol{w}\|^2 + \sum_i \left(y_i - \boldsymbol{w}^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x_i})\right)^2$$

  - $\|\boldsymbol{w}\|^2$ is a regularisation term
  - The weights must lie in the space spanned by the set of extended feature vectors $\{\boldsymbol{\phi}(\boldsymbol{x}_k)|k=1,2,\ldots,m\}$
  - Thus we can write

$$\boldsymbol{w} = \sum_i \alpha_i\,\boldsymbol{\phi}(\boldsymbol{x}_i)$$

    - Note that here α_i are just parameters; they are not Lagrange multipliers and they can be negative
  - Substituting this into the objective function for ridge regression we get a quadratic optimisation problem in $\boldsymbol{\alpha}$ that just depends on the inner products $\boldsymbol{\phi}^{\mathsf{T}}(\boldsymbol{x}_i)\,\boldsymbol{\phi}(\boldsymbol{x}_j)$
  - We can use the kernel trick
- *Kernel PCA*
  - For kernel PCA we map features into an external feature space
  - We then use the dual form of PCA (which we've done in an earlier lecture)
  - This allows us to find non-linear manifolds where the data varies
- *Kernel Canonical Correlation Analysis*

- Canonical correlation analysis finds correlations between datasets
- The linear form is a bit naff
- But the kernel form can give nice results

- *Gaussian Processes*
  - Gaussian Processes also use kernels
  - They are a bit different to other kernel methods
    * we don't think of the working in an extended feature space
    * but they are PSD
  - They are one of the most successful methods for doing regression
  - We will look at them later

3 Exercises

3.1 Quadratic Kernels

- Show that the kernel function $K(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{\phi}^{\mathsf{T}}(\boldsymbol{x})\,\boldsymbol{\phi}(\boldsymbol{y})$, where

$$\boldsymbol{\phi}(\boldsymbol{x}) = (x_1^2, x_2^2, x_3^2, \sqrt{2}\,x_1x_2, \sqrt{2}\,x_1x_3, \sqrt{2}\,x_2x_3)$$

  can be written as $(\boldsymbol{x}^{\mathsf{T}}\boldsymbol{y})^2$ is $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors of length 3.

- Answer below

3.2 Kernel Ridge Regression

- Work out the details for kernel ridge regression
- Have a go at implementing kernel ridge regression on a real data set
- I'll leave you to work this out

4 Experiments

4.1 Gram Matrix

- Generate ten random vectors $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{10})$ where $\boldsymbol{x}_k \in \mathbb{R}^5$
- Compute the Gram matrix $\mathbf{K}$ with components

$$K_{kl} = K(\boldsymbol{x}_k, \boldsymbol{x}_l) = \mathrm{e}^{-\|\boldsymbol{x}_k - \boldsymbol{x}_l\|^2}$$

- Show that $\mathbf{K}$ is positive definite by computing its eigenvalues

```
n = 10;
X=randn(n,5);                  % matrix of vectors
K = zeros(n,n);                 % define holder for Gram
for i = 1:n
  x = X(i,:);
  for j = 1:n
    y = X(j,:);
    K(i,j) = exp(-norm(x-y)^2); % define elements of Gram matrix
  endfor
endfor

K            % Gram matrix
eig(K)     % Eigenvalues should all be non-negative
```

5 Answers

5.1 Quadratic Kernel

- This is just straightforward algebra

$$\begin{aligned}\boldsymbol{\phi}^{\mathsf{T}}(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{y}) &= x_1^2 y_1^2 + x_2^2 y_2^2 + x_3^2 y_3^2 + 2 x_1 x_2 y_1 y_2 + 2\, x_1 x_3 y_1 y_3 + 2 x_2 x_3 y_2 y_3 \\ &= (x_1 y_1 + x_2 y_2 + x_3 y_2)^2 = (\boldsymbol{x}^{\mathsf{T}}\boldsymbol{y})^2\end{aligned}$$

- In the lecture notes we did the 2-d case

- Note that the more general polynomial kernel is

$$K_p(\boldsymbol{x}, \boldsymbol{y}) = (1 + \boldsymbol{x}^{\mathsf{T}}\boldsymbol{y})^p$$

  - this is more commonly used as it incorporates the lower dimensional polynomial kernels