

The Representer Theorem and Kernelization

David S. Rosenberg

New York University

February 26, 2019

Contents

- 1 Solutions in the “span of the data,” and so what?
- 2 Math Review: Inner Product Spaces and Projections (Hilbert Spaces)
- 3 The Representer Theorem
- 4 Reparameterizing our Generalized Objective Function
- 5 Kernel Ridge Regression
- 6 Kernel SVM
- 7 Are we done yet?

Solutions in the “span of the data,” and so what?

SVM solution is in the “span of the data”

- We found the SVM dual problem can be written as:

$$\begin{aligned} \sup_{\alpha \in \mathbf{R}^n} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_j^T x_i \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i = 0 \\ & \alpha_i \in \left[0, \frac{C}{n}\right] \quad i = 1, \dots, n. \end{aligned}$$

- Given solution α^* to dual, primal solution is $w^* = \sum_{i=1}^n \alpha_i^* y_i x_i$.
- Notice: w^* is a linear combination of training inputs x_1, \dots, x_n .
- We refer to this phenomenon by saying “ w^* is in the **span of the data.**”
 - Or in math, $w^* \in \text{span}(x_1, \dots, x_n)$.

Ridge regression solution is in the “span of the data”

- The ridge regression solution for regularization parameter $\lambda > 0$ is

$$w^* = \arg \min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^n \{w^T x_i - y_i\}^2 + \lambda \|w\|_2^2.$$

- This has a closed form solution (Homework #4):

$$w^* = (X^T X + \lambda I)^{-1} X^T y,$$

where X is the design matrix, with x_1, \dots, x_n as rows.

Ridge regression solution is in the “span of the data”

- Rearranging $w^* = (X^T X + \lambda I)^{-1} X^T y$, we can show that (also Homework #4):

$$\begin{aligned} w^* &= X^T \underbrace{\left(\frac{1}{\lambda} y - \frac{1}{\lambda} X w^* \right)}_{\alpha^*} \\ &= X^T \alpha^* = \sum_{i=1}^n \alpha_i^* x_i. \end{aligned}$$

- So w^* is in the span of the data.
 - i.e. $w^* \in \text{span}(x_1, \dots, x_n)$

If solution is in the span of the data, we can reparameterize

- The ridge regression solution for regularization parameter $\lambda > 0$ is

$$w^* = \arg \min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^n \{w^T x_i - y_i\}^2 + \lambda \|w\|_2^2.$$

- We now know that $w^* \in \text{span}(x_1, \dots, x_n) \subset \mathbf{R}^d$.
- So rather than minimizing over all of \mathbf{R}^d , we can minimize over $\text{span}(x_1, \dots, x_n)$.

$$w^* = \arg \min_{w \in \text{span}(x_1, \dots, x_n)} \frac{1}{n} \sum_{i=1}^n \{w^T x_i - y_i\}^2 + \lambda \|w\|_2^2.$$

- How can we conveniently write an optimization problem over the span of some vectors?

If solution is in the span of the data, we can reparameterize

- Note that for any $w \in \text{span}(x_1, \dots, x_n)$, we have $w = X^T \alpha$, for some $\alpha \in \mathbf{R}^n$.
- So let's replace w with $X^T \alpha$ in our optimization problem:

$$\text{[original]} \quad w^* = \arg \min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^n \{w^T x_i - y_i\}^2 + \lambda \|w\|_2^2$$

$$\text{[reparameterized]} \quad \alpha^* = \arg \min_{\alpha \in \mathbf{R}^n} \frac{1}{n} \sum_{i=1}^n \{(X^T \alpha)^T x_i - y_i\}^2 + \lambda \|X^T \alpha\|_2^2.$$

- To get w^* from the reparameterized optimization problem, we just take $w^* = X^T \alpha^*$.
- We changed the dimension of our optimization variable from d to n . Is this useful?

Consider very large feature spaces

- Suppose we have a 300-million dimension feature space [very large]
 - (e.g. using high order monomial interaction terms as features, as described last lecture)
- Suppose we have a training set of 300,000 examples [fairly large]
- In the original formulation, we solve a 300-million dimension optimization problem.
- In the reparameterized formulation, we solve a 300,000-dimension optimization problem.
- **This is why we care** about when the solution is in the span of the data.
- This reparameterization is interesting when we have more features than data ($d \gg n$).

What's next?

- For SVM and ridge regression, we found that the solution is in the span of the data.
 - derived in two rather ad-hoc ways
- Up next: The Representer Theorem, which shows that this “span of the data” result occurs far more generally, and we prove it using basic linear algebra.

Math Review: Inner Product Spaces and Projections (Hilbert Spaces)

Inner Product Space (or “Pre-Hilbert” Spaces)

An **inner product space** (over reals) is a vector space \mathcal{V} and an **inner product**, which is a mapping

$$\langle \cdot, \cdot \rangle : \mathcal{V} \times \mathcal{V} \rightarrow \mathbf{R}$$

that has the following properties $\forall x, y, z \in \mathcal{V}$ and $a, b \in \mathbf{R}$:

- Symmetry: $\langle x, y \rangle = \langle y, x \rangle$
- Linearity: $\langle ax + by, z \rangle = a \langle x, z \rangle + b \langle y, z \rangle$
- Positive-definiteness: $\langle x, x \rangle \geq 0$ and $\langle x, x \rangle = 0 \iff x = 0$.

Norm from Inner Product

For an inner product space, we define a norm as

$$\|x\| = \sqrt{\langle x, x \rangle}.$$

Example

\mathbf{R}^d with standard Euclidean inner product is an inner product space:

$$\langle x, y \rangle := x^T y \quad \forall x, y \in \mathbf{R}^d.$$

Norm is

$$\|x\| = \sqrt{x^T x}.$$

What norms can we get from an inner product?

Theorem (Parallelogram Law)

A norm $\|\cdot\|$ can be written in terms of an inner product on \mathcal{V} iff $\forall x, x' \in \mathcal{V}$

$$2\|x\|^2 + 2\|x'\|^2 = \|x + x'\|^2 + \|x - x'\|^2,$$

and if it can, the inner product is given by the **polarization identity**

$$\langle x, x' \rangle = \frac{\|x\|^2 + \|x'\|^2 - \|x - x'\|^2}{2}.$$

Example

ℓ_1 norm on \mathbf{R}^d is NOT generated by an inner product. [Exercise]

Is ℓ_2 norm on \mathbf{R}^d generated by an inner product?

Orthogonality (Definitions)

Definition

Two vectors are **orthogonal** if $\langle x, x' \rangle = 0$. We denote this by $x \perp x'$.

Definition

x is orthogonal to a set S , i.e. $x \perp S$, if $x \perp s$ for all $s \in S$.

Pythagorean Theorem

Theorem (Pythagorean Theorem)

If $x \perp x'$, then $\|x + x'\|^2 = \|x\|^2 + \|x'\|^2$.

Proof.

We have

$$\begin{aligned}\|x + x'\|^2 &= \langle x + x', x + x' \rangle \\ &= \langle x, x \rangle + \langle x, x' \rangle + \langle x', x \rangle + \langle x', x' \rangle \\ &= \|x\|^2 + \|x'\|^2.\end{aligned}$$



Projection onto a Plane (Rough Definition)

- Choose some $x \in \mathcal{V}$.
- Let M be a subspace of inner product space \mathcal{V} .
- Then m_0 is the **projection of x onto M** ,
 - if $m_0 \in M$ and is the closest point to x in M .
- In math: For all $m \in M$,

$$\|x - m_0\| \leq \|x - m\|.$$

Hilbert Space

- Projections exist for all finite-dimensional inner product spaces.
- We want to allow infinite-dimensional spaces.
- Need an extra condition called **completeness**.
- A space is **complete** if all Cauchy sequences in the space converge.

Definition

A **Hilbert space** is a complete inner product space.

Example

Any finite dimensional inner product space is a Hilbert space.

The Projection Theorem

Theorem (Classical Projection Theorem)

- \mathcal{H} a Hilbert space
- M a closed subspace of \mathcal{H} (picture a hyperplane through the origin)
- For any $x \in \mathcal{H}$, there **exists a unique** $m_0 \in M$ for which

$$\|x - m_0\| \leq \|x - m\| \quad \forall m \in M.$$

- This m_0 is called the **[orthogonal] projection of x onto M** .
- Furthermore, $m_0 \in M$ is the projection of x onto M iff

$$x - m_0 \perp M.$$

Projection Reduces Norm

Theorem

Let M be a closed subspace of \mathcal{H} . For any $x \in \mathcal{H}$, let $m_0 = \text{Proj}_M x$ be the projection of x onto M . Then

$$\|m_0\| \leq \|x\|,$$

with equality only when $m_0 = x$.

Proof.

$$\begin{aligned}\|x\|^2 &= \|m_0 + (x - m_0)\|^2 \text{ (note: } x - m_0 \perp m_0 \text{ by Projection theorem)} \\ &= \|m_0\|^2 + \|x - m_0\|^2 \text{ by Pythagorean theorem} \\ \|m_0\|^2 &= \|x\|^2 - \|x - m_0\|^2\end{aligned}$$

Then $\|x - m_0\|^2 \geq 0$ implies $\|m_0\|^2 \leq \|x\|^2$. If $\|x - m_0\|^2 = 0$, then $x = m_0$, by definition of norm. □

The Representer Theorem

Generalize from SVM Objective

- SVM objective:

$$\min_{w \in \mathbf{R}^d} \frac{1}{2} \|w\|^2 + \frac{c}{n} \sum_{i=1}^n \max(0, 1 - y_i [\langle w, x_i \rangle]).$$

- Generalized objective:

$$\min_{w \in \mathcal{H}} R(\|w\|) + L(\langle w, x_1 \rangle, \dots, \langle w, x_n \rangle),$$

where

- $R: [0, \infty) \rightarrow \mathbf{R}$ is nondecreasing (**Regularization term**)
- and $L: \mathbf{R}^n \rightarrow \mathbf{R}$ is arbitrary. (**Loss term**)

General Objective Function for Linear Hypothesis Space (Details)

- Generalized objective:

$$\min_{w \in \mathcal{H}} R(\|w\|) + L(\langle w, x_1 \rangle, \dots, \langle w, x_n \rangle),$$

where

- $w, x_1, \dots, x_n \in \mathcal{H}$ for some Hilbert space \mathcal{H} . (We typically have $\mathcal{H} = \mathbf{R}^d$.)
- $\|\cdot\|$ is the norm corresponding to the inner product of \mathcal{H} . (i.e. $\|w\| = \sqrt{\langle w, w \rangle}$)
- $R: [0, \infty) \rightarrow \mathbf{R}$ is nondecreasing (**Regularization term**), and
- $L: \mathbf{R}^n \rightarrow \mathbf{R}$ is arbitrary (**Loss term**).

General Objective Function for Linear Hypothesis Space (Details)

- **Generalized objective:**

$$\min_{w \in \mathcal{H}} R(\|w\|) + L(\langle w, x_1 \rangle, \dots, \langle w, x_n \rangle)$$

- What's "linear"?
- The prediction/score function $x \mapsto \langle w, x \rangle$ is linear – in what?
 - in parameter vector w , and
 - in the feature vector x .
- Why? [Real-valued] inner products are linear in each argument.
- **The important part is the linearity in the parameter w .**

General Objective Function for Linear Hypothesis Space (Details)

- **Generalized objective:**

$$\min_{w \in \mathcal{H}} R(\|w\|) + L(\langle w, x_1 \rangle, \dots, \langle w, x_n \rangle)$$

- Ridge regression and SVM are of this form. (Verify this!)
- What if we penalize with $\lambda\|w\|_2$ instead of $\lambda\|w\|_2^2$? Yes!.
- What if we use lasso regression? No! ℓ_1 norm does not correspond to an inner product.

The Representer Theorem: Quick Summary

- **Generalized objective:**

$$w^* = \arg \min_{w \in \mathcal{H}} R(\|w\|) + L(\langle w, x_1 \rangle, \dots, \langle w, x_n \rangle)$$

- Representer theorem tells us we can look for w^* in the span of the data:

$$w^* = \arg \min_{w \in \text{span}(x_1, \dots, x_n)} R(\|w\|) + L(\langle w, x_1 \rangle, \dots, \langle w, x_n \rangle).$$

- So we can reparameterize as before:

$$\alpha^* = \arg \min_{\alpha \in \mathbb{R}^n} R\left(\left\|\sum_{i=1}^n \alpha_i x_i\right\|\right) + L\left(\left\langle \sum_{i=1}^n \alpha_i x_i, x_1 \right\rangle, \dots, \left\langle \sum_{i=1}^n \alpha_i x_i, x_n \right\rangle\right).$$

- Our reparameterization trick applies much more broadly than SVM and ridge.

The Representer Theorem

Theorem (Representer Theorem)

Let

$$J(w) = R(\|w\|) + L(\langle w, x_1 \rangle, \dots, \langle w, x_n \rangle),$$

where

- $w, x_1, \dots, x_n \in \mathcal{H}$ for some Hilbert space \mathcal{H} . (We typically have $\mathcal{H} = \mathbf{R}^d$.)
- $\|\cdot\|$ is the norm corresponding to the inner product of \mathcal{H} . (i.e. $\|w\| = \sqrt{\langle w, w \rangle}$)
- $R: [0, \infty) \rightarrow \mathbf{R}$ is nondecreasing (**Regularization term**), and
- $L: \mathbf{R}^n \rightarrow \mathbf{R}$ is arbitrary (**Loss term**).

Then

- If $M = \text{span}(x_1, \dots, x_n)$, then $J(\text{Proj}_M w) \leq J(w)$ for any $w \in \mathcal{H}$.
- If $J(w)$ has a minimizer, then it **has a minimizer of the form** $w^* = \sum_{i=1}^n \alpha_i x_i$.
- If R is strictly increasing, then all minimizers have this form. (Proof in homework.)

The Representer Theorem (Proof)

- 1 Fix any $w \in \mathcal{H}$.
- 2 Let $w_M = \text{Proj}_M w$.
- 3 Residual $w - w_M$ is orthogonal to x for all $x \in M$.
- 4 $\langle w, x_i \rangle = \langle w_M + w - w_M, x_i \rangle = \langle w_M, x_i \rangle + \langle w - w_M, x_i \rangle = \langle w_M, x_i \rangle \quad \forall i$.
- 5 $L(\langle w, x_1 \rangle, \dots, \langle w, x_n \rangle) = L(\langle w_M, x_1 \rangle, \dots, \langle w_M, x_n \rangle)$.
- 6 Projections decrease norms $\implies \|w_M\| \leq \|w\|$.
- 7 Since R is nondecreasing, $R(\|w_M\|) \leq R(\|w\|)$.
- 8 $J(w_M) \leq J(w)$. [Proves first result.]
- 9 If w^* minimizes $J(w)$, then $w_M^* = \text{Proj}_M w^*$ is also a minimizer, since $J(w_M^*) \leq J(w^*)$.
- 10 So $\exists \alpha$ s.t. $w_M^* = \sum_{i=1}^n \alpha_i x_i$ is a minimizer of $J(w)$.

Q.E.D.

Sufficient Condition for Existence of a Minimizer

Theorem

^aLet

$$J(w) = R(\|w\|) + L(\langle w, x_1 \rangle, \dots, \langle w, x_n \rangle),$$

and let $M = \text{span}(x_1, \dots, x_n)$. Then under the same conditions given in the Representer theorem, if w_M^* minimizes $J(w)$ **over the set M** , then w_M^* minimizes $J(w)$ over all \mathcal{H} .

^aThanks to [Mingsi Long](#) for suggesting this nice theorem and proof.

- One consequence of the Representer theorem only applies if $J(w)$ has a minimizer over \mathcal{H} . This theorem tells us that it's sufficient to check for a constrained minimizer of $J(w)$ over M . If one exists, then it's also an unconstrained minimizer of $J(w)$ over \mathcal{H} . If there is no constrained minimizer over M , then $J(w)$ has no minimizer over \mathcal{H} (by the Representer theorem).
- Bottom Line: We can jump straight to minimizing over M , the “span of the data”.

Sufficient Condition for Existence of a Minimizer (Proof)

- 1 Let $w_M^* \in \arg \min_{w \in M} J(w)$. [the constrained minimizer]
- 2 Consider any $w \in \mathcal{H}$.
- 3 Let $w_M = \text{Proj}_M w$.
- 4 By the Representer theorem, $J(w_M) \leq J(w)$.
- 5 $J(w_M^*) \leq J(w_M)$ by definition of w_M^* .
- 6 Thus for any $w \in \mathcal{H}$, $J(w_M^*) \leq J(w)$.
- 7 Therefore w_M^* minimizes $J(w)$ over \mathcal{H}

QED

Reparameterizing our Generalized Objective Function

Rewriting the Objective Function

- Define the training score function $s : \mathbf{R}^d \rightarrow \mathbf{R}^n$ by

$$s(w) = \begin{pmatrix} \langle w, x_1 \rangle \\ \vdots \\ \langle w, x_n \rangle \end{pmatrix},$$

which gives the **training score vector** for any w .

- We can then rewrite the objective function as

$$J(w) = R(\|w\|) + L(s(w)),$$

where now $L : \mathbf{R}^{n \times 1} \rightarrow \mathbf{R}$ takes a column vector as input.

- This will allow us to have a slick reparameterized version...

Reparameterize the Generalized Objective

- By the Representer Theorem, it's sufficient to minimize $J(w)$ for w of the form $\sum_{i=1}^n \alpha_i x_i$.
- Plugging this form into $J(w)$, we see we can just minimize

$$J_0(\alpha) = R\left(\left\|\sum_{i=1}^n \alpha_i x_i\right\|\right) + L\left(s\left(\sum_{i=1}^n \alpha_i x_i\right)\right)$$

over $\alpha = (\alpha_1, \dots, \alpha_n)^T \in \mathbf{R}^{n \times 1}$.

- With some new notation, we can substantially simplify
 - the norm piece $\|w\| = \left\|\sum_{i=1}^n \alpha_i x_i\right\|$, and
 - the score piece $s(w) = s\left(\sum_{i=1}^n \alpha_i x_i\right)$.

Simplifying the Reparameterized Norm

- For the norm piece $\|w\| = \|\sum_{i=1}^n \alpha_i x_i\|$, we have

$$\begin{aligned}\|w\|^2 &= \langle w, w \rangle \\ &= \left\langle \sum_{i=1}^n \alpha_i x_i, \sum_{j=1}^n \alpha_j x_j \right\rangle \\ &= \sum_{i,j=1}^n \alpha_i \alpha_j \langle x_i, x_j \rangle.\end{aligned}$$

- This expression involves the n^2 inner products between all pairs of input vectors.
- We often put those values together into a matrix...

The Gram Matrix

Definition

The **Gram matrix** of a set of points x_1, \dots, x_n in an inner product space is defined as

$$K = (\langle x_i, x_j \rangle)_{i,j} = \begin{pmatrix} \langle x_1, x_1 \rangle & \cdots & \langle x_1, x_n \rangle \\ \vdots & \ddots & \cdots \\ \langle x_n, x_1 \rangle & \cdots & \langle x_n, x_n \rangle \end{pmatrix}.$$

- This is the traditional definition from linear algebra.
- Later today we'll introduce the notion of a “kernel matrix”
 - The Gram matrix is a special case of a **kernel matrix** for the identity feature map.
 - That's why we write K for the Gram matrix instead of G , as done elsewhere.
- NOTE: In ML, we often use Gram matrix and kernel matrix to mean the same thing. Don't get too hung up on the definitions.

Example: Gram Matrix for the Dot Product

- Consider $x_1, \dots, x_n \in \mathbf{R}^{d \times 1}$ with the standard inner product $\langle x, x' \rangle = x^T x'$.
- Let $X \in \mathbf{R}^{n \times d}$ be the **design matrix**, which has each input vector as a row:

$$X = \begin{pmatrix} -x_1^T - \\ \vdots \\ -x_n^T - \end{pmatrix}.$$

- Then the Gram matrix is

$$\begin{aligned} K &= \begin{pmatrix} x_1^T x_1 & \cdots & x_1^T x_n \\ \vdots & \ddots & \vdots \\ x_n^T x_1 & \cdots & x_n^T x_n \end{pmatrix} = \begin{pmatrix} -x_1^T - \\ \vdots \\ -x_n^T - \end{pmatrix} \begin{pmatrix} | & \cdots & | \\ x_1 & \cdots & x_n \\ | & \cdots & | \end{pmatrix} \\ &= XX^T \end{aligned}$$

Simplifying the Reparametrized Norm

- With $w = \sum_{i=1}^n \alpha_i x_i$, we have

$$\begin{aligned}\|w\|^2 &= \langle w, w \rangle \\ &= \left\langle \sum_{i=1}^n \alpha_i x_i, \sum_{j=1}^n \alpha_j x_j \right\rangle \\ &= \sum_{i,j=1}^n \alpha_i \alpha_j \langle x_i, x_j \rangle \\ &= \alpha^T K \alpha.\end{aligned}$$

Simplifying the Training Score Vector

- The score for x_j for $w = \sum_{i=1}^n \alpha_i x_i$ is

$$\langle w, x_j \rangle = \left\langle \sum_{i=1}^n \alpha_i x_i, x_j \right\rangle = \sum_{i=1}^n \alpha_i \langle x_i, x_j \rangle$$

- The training score vector is

$$\begin{aligned} s \left(\sum_{i=1}^n \alpha_i x_i \right) &= \begin{pmatrix} \sum_{i=1}^n \alpha_i \langle x_i, x_1 \rangle \\ \vdots \\ \sum_{i=1}^n \alpha_i \langle x_i, x_n \rangle \end{pmatrix} = \begin{pmatrix} \alpha_1 \langle x_1, x_1 \rangle + \cdots + \alpha_n \langle x_n, x_1 \rangle \\ \vdots \\ \alpha_1 \langle x_1, x_n \rangle + \cdots + \alpha_n \langle x_n, x_n \rangle \end{pmatrix} \\ &= \begin{pmatrix} \langle x_1, x_1 \rangle & \cdots & \langle x_1, x_n \rangle \\ \vdots & \ddots & \cdots \\ \langle x_n, x_1 \rangle & \cdots & \langle x_n, x_n \rangle \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} \\ &= K \alpha \end{aligned}$$

Reparameterized Objective

- Putting it all together, our reparameterized objective function can be written as

$$\begin{aligned} J_0(\alpha) &= R\left(\left\|\sum_{i=1}^n \alpha_i x_i\right\|\right) + L\left(s\left(\sum_{i=1}^n \alpha_i x_i\right)\right) \\ &= R\left(\sqrt{\alpha^T K \alpha}\right) + L(K\alpha), \end{aligned}$$

which we minimize over $\alpha \in \mathbf{R}^n$.

- All information** needed about x_1, \dots, x_n is summarized in the Gram matrix K .
- We're now minimizing over \mathbf{R}^n rather than \mathbf{R}^d .
- If $d \gg n$, this can be a big win computationally (at least once K is computed).

Reparameterizing Predictions

- Suppose we've found

$$\alpha^* \in \arg \min_{\alpha \in \mathbf{R}^n} R\left(\sqrt{\alpha^T K \alpha}\right) + L(K\alpha).$$

- Then we know $w^* = \sum_{i=1}^n \alpha_i^* x_i$ is a solution to

$$\arg \min_{w \in \mathcal{H}} R(\|w\|) + L(\langle w, x_1 \rangle, \dots, \langle w, x_n \rangle).$$

- The prediction on a new point $x \in \mathcal{H}$ is

$$\hat{f}(x) = \langle w^*, x \rangle = \sum_{i=1}^n \alpha_i^* \langle x_i, x \rangle.$$

- To make a new prediction, we may need to touch all the training inputs x_1, \dots, x_n .

- It will be convenient to define the following column vector for any $x \in \mathcal{H}$:

$$k_x = \begin{pmatrix} \langle x_1, x \rangle \\ \vdots \\ \langle x_n, x \rangle \end{pmatrix}$$

- Then we can write our predictions on a new point x as

$$\hat{f}(x) = k_x^T \alpha^*$$

Summary So Far

- Original plan:
 - Find $w^* \in \arg \min_{w \in \mathcal{H}} R(\|w\|) + L(\langle w, x_1 \rangle, \dots, \langle w, x_n \rangle)$
 - Predict with $\hat{f}(x) = \langle w^*, x \rangle$.
- We showed that the following is equivalent:
 - Find $\alpha^* \in \arg \min_{\alpha \in \mathbb{R}^n} R(\sqrt{\alpha^T K \alpha}) + L(K\alpha)$
 - Predict with $\hat{f}(x) = k_x^T \alpha^*$, where

$$K = \begin{pmatrix} \langle x_1, x_1 \rangle & \cdots & \langle x_1, x_n \rangle \\ \vdots & \ddots & \dots \\ \langle x_n, x_1 \rangle & \cdots & \langle x_n, x_n \rangle \end{pmatrix} \quad \text{and} \quad k_x = \begin{pmatrix} \langle x_1, x \rangle \\ \vdots \\ \langle x_n, x \rangle \end{pmatrix}$$

- Every element $x \in \mathcal{H}$ occurs inside an inner products with a training input $x_i \in \mathcal{H}$.

Definition

A method is **kernelized** if every feature vector $\psi(x)$ only appears inside an inner product with another feature vector $\psi(x')$. This applies to both the optimization problem and the prediction function.

- Here we are using $\psi(x) = x$. Thus finding

$$\alpha^* \in \arg \min_{\alpha \in \mathbf{R}^n} R\left(\sqrt{\alpha^T K \alpha}\right) + L(K\alpha)$$

and making predictions with $\hat{f}(x) = k_x^T \alpha^*$ is a **kernelization** of finding

$$w^* \in \arg \min_{w \in \mathcal{H}} R(\|w\|) + L(\langle w, x_1 \rangle, \dots, \langle w, x_n \rangle)$$

and making predictions with $\hat{f}(x) = \langle w^*, x \rangle$.

How to kernelize?

- Our principle tool for kernelization is reparameterization by the representer theorem.
- There are other methods – we used duality for SVM and bare hands for ridge regression.
- Below, we highlight key differences between
 - kernelized ridge regression and kernelized SVM at prediction time..

Kernel Ridge Regression

Kernelizing Ridge Regression

- Ridge Regression:

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \|Xw - y\|^2 + \lambda \|w\|^2$$

- Plugging in $w = \sum_{i=1}^n \alpha_i x_i$, we get the kernelized ridge regression objective function:

$$\min_{\alpha \in \mathbf{R}^n} \frac{1}{n} \|K\alpha - y\|^2 + \lambda \alpha^T K \alpha$$

- This is usually just called **kernel ridge regression**.

Kernel Ridge Regression Solutions

- For $\lambda > 0$, the **ridge regression solution** is

$$w^* = (X^T X + \lambda I)^{-1} X^T y$$

- and the **kernel ridge regression solution** is

$$\begin{aligned} \alpha^* &= (X X^T + \lambda I)^{-1} y \\ &= (K + \lambda I)^{-1} y \end{aligned}$$

- (Shown in homework.)
- For ridge regression we're dealing with a $d \times d$ matrix.
- For kernel ridge regression we're dealing an $n \times n$ matrix.

- Predictions in terms of w^* :

$$\hat{f}(x) = x^T w^*$$

- Predictions in terms of α^* :

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

- For kernel ridge regression, need to access all training inputs x_1, \dots, x_n to predict.
- For SVM, we may not...

Kernel SVM

Kernelized SVM (From Representer Theorem)

- The SVM objective:

$$\min_{w \in \mathbf{R}^d} \frac{1}{2} \|w\|^2 + \frac{c}{n} \sum_{i=1}^n \max(0, 1 - y_i w^T x_i).$$

- Plugging in $w = \sum_{i=1}^n \alpha_i x_i$, we get

$$\min_{\alpha \in \mathbf{R}^n} \frac{1}{2} \alpha^T K \alpha + \frac{c}{n} \sum_{i=1}^n \max(0, 1 - y_i (K \alpha)_i)$$

- Predictions with

$$\hat{f}(x) = x^T w^* = \sum_{i=1}^n \alpha_i^* x_i^T x.$$

- This is one way to kernelize SVM...

Kernelized SVM (From Lagrangian Duality)

- Kernelized SVM from computing the Lagrangian Dual Problem:

$$\begin{aligned} \max_{\alpha \in \mathbf{R}^n} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_j^T x_i \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i = 0 \\ & \alpha_i \in \left[0, \frac{C}{n}\right] \quad i = 1, \dots, n. \end{aligned}$$

- If α^* is an optimal value, then

$$w^* = \sum_{i=1}^n \alpha_i^* y_i x_i \quad \text{and} \quad \hat{f}(x) = \sum_{i=1}^n \alpha_i^* y_i x_i^T x.$$

- Note that the prediction function is also kernelized.

Sparsity in the Data from Complementary Slackness

- Kernelized predictions given by

$$\hat{f}(x) = \sum_{i=1}^n \alpha_i^* y_i x_i^T x.$$

- By a Lagrangian duality analysis (specifically from complementary slackness), we find

$$y_i \hat{f}(x_i) < 1 \implies \alpha_i^* = \frac{c}{n}$$

$$y_i \hat{f}(x_i) = 1 \implies \alpha_i^* \in \left[0, \frac{c}{n}\right]$$

$$y_i \hat{f}(x_i) > 1 \implies \alpha_i^* = 0$$

- So we can leave out any x_i “on the good side of the margin” ($y_i \hat{f}(x_i) > 1$).
- x_i 's that we must keep, because $\alpha_i^* \neq 0$, are called **support vectors**.

Are we done yet?

Computational considerations – we're not really done yet

- Suppose our feature space is $\mathcal{H} = \mathbf{R}^d$.
- And we use representer theorem to kernelize.
- Get optimization problem over \mathbf{R}^n rather than over \mathbf{R}^d :

$$\text{[original]} \quad w^* = \arg \min_{w \in \mathbf{R}^d} R(\|w\|) + L(\langle w, x_1 \rangle, \dots, \langle w, x_n \rangle)$$

$$\text{[kernelized]} \quad \alpha^* = \arg \min_{\alpha \in \mathbf{R}^n} R\left(\sqrt{\alpha^T K \alpha}\right) + L(K\alpha)$$

- This seems like a good move if $d \gg n$.
- However, there is still a hidden dependence on d in the kernelized form – do you see it?

Computational considerations – we're not really done yet

- Get optimization problem over \mathbf{R}^n rather than over \mathbf{R}^d :

$$\text{[original]} \quad w^* = \arg \min_{w \in \mathbf{R}^d} R(\|w\|) + L(\langle w, x_1 \rangle, \dots, \langle w, x_n \rangle)$$

$$\text{[kernelized]} \quad \alpha^* = \arg \min_{\alpha \in \mathbf{R}^n} R\left(\sqrt{\alpha^T K \alpha}\right) + L(K\alpha)$$

- For the standard inner product, $K_{ij} = \langle x_i, x_j \rangle = x_i^T x_j$, where $x_i, x_j \in \mathbf{R}^d$.
- This is still $O(d)$, and can be too slow for huge feature spaces.
- The essence of the “**kernel trick**” is getting around this $O(d)$ dependence.