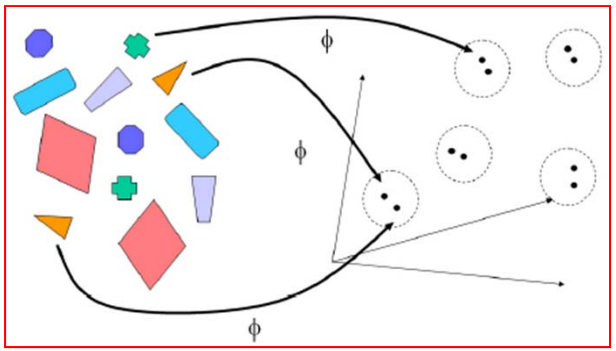
Advanced Introduction to Machine Learning

10715, Fall 2014

The Kernel Trick, Reproducing Kernel Hilbert Space, and the Representer Theorem

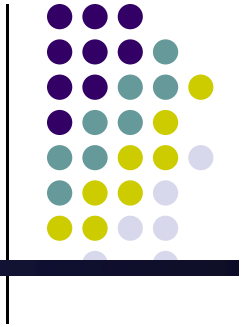


Eric Xing

Lecture 6, September 24, 2014

Reading:

Recap: the SVM problem



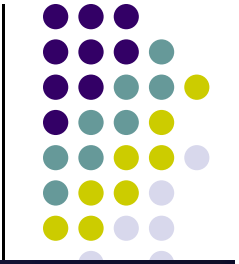
- We solve the following constrained opt problem:

$$\max_{\alpha} \quad \mathcal{J}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

$$\text{s.t.} \quad 0 \le \alpha_i \le C, \quad i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0.$$

- This is a **quadratic programming** problem.
  - A global maximum of α_i can always be found.
  - The solution: $w = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$
  - How to predict: $\mathbf{w}^T \mathbf{x}_{\text{new}} + b \lessgtr 0$

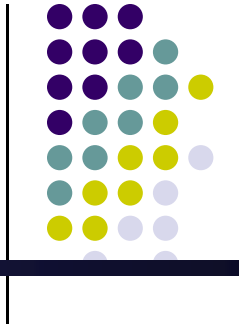


$$\max_{\alpha} \mathcal{J}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

$$\mathbf{w}^T \mathbf{x}_{\text{new}} + b \lessgtr 0$$

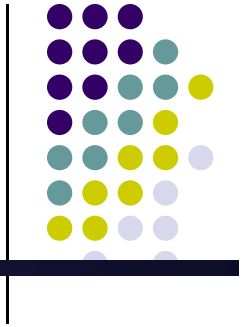
- Kernel
- Point rule or average rule
- Can we predict vec(y)?

Outline



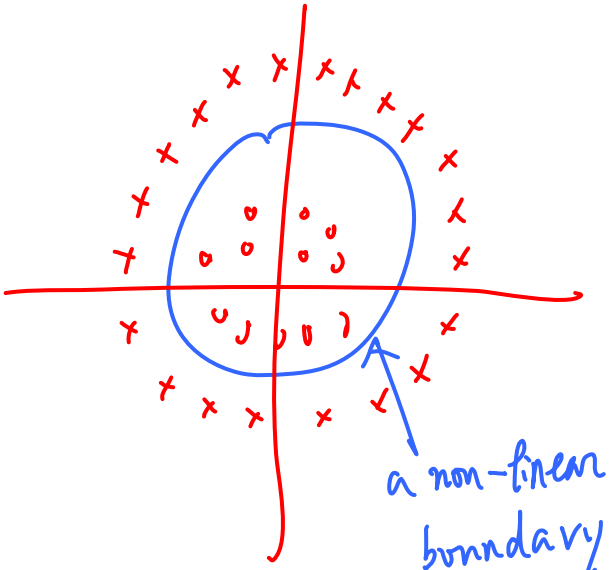
- The Kernel trick
- Maximum entropy discrimination
- Structured SVM, aka, Maximum Margin Markov Networks

(1) Non-linear Decision Boundary



- So far, we have only considered large-margin classifier with a linear decision boundary
- How to generalize it to become nonlinear?
- Key idea: transform $\mathbf{x}_i$ to a higher dimensional space to "make life easier"
  - Input space: the space the point $\mathbf{x}_i$ are located
  - Feature space: the space of $\phi(\mathbf{x}_i)$ after transformation
- Why transform?
  - Linear operation in the feature space is equivalent to non-linear operation in input space
  - Classification can become easier with a proper transformation. In the XOR problem, for example, adding a new feature of x_1x_2 make the problem linearly separable (homework)

Non-linear Decision Boundary



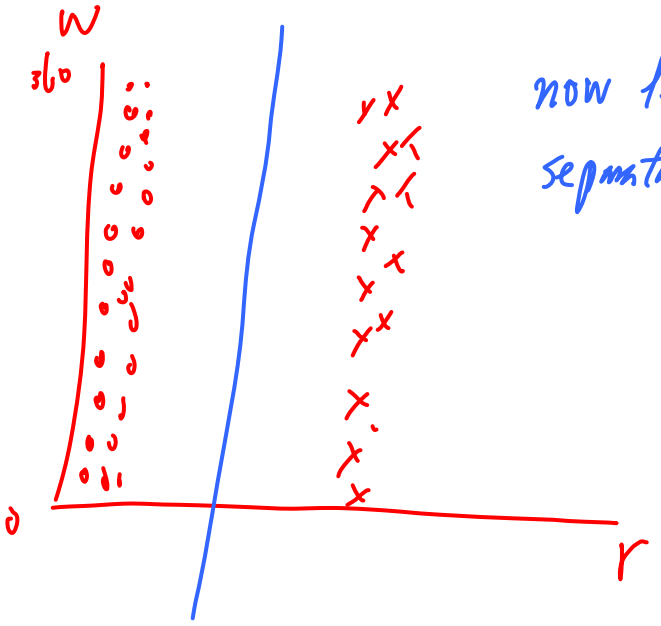
this data set is not linearly separable!

transformation

$(x_1, x_2) \rightarrow (r(x_1, x_2), w(x_1, x_2))$

radius, angle.

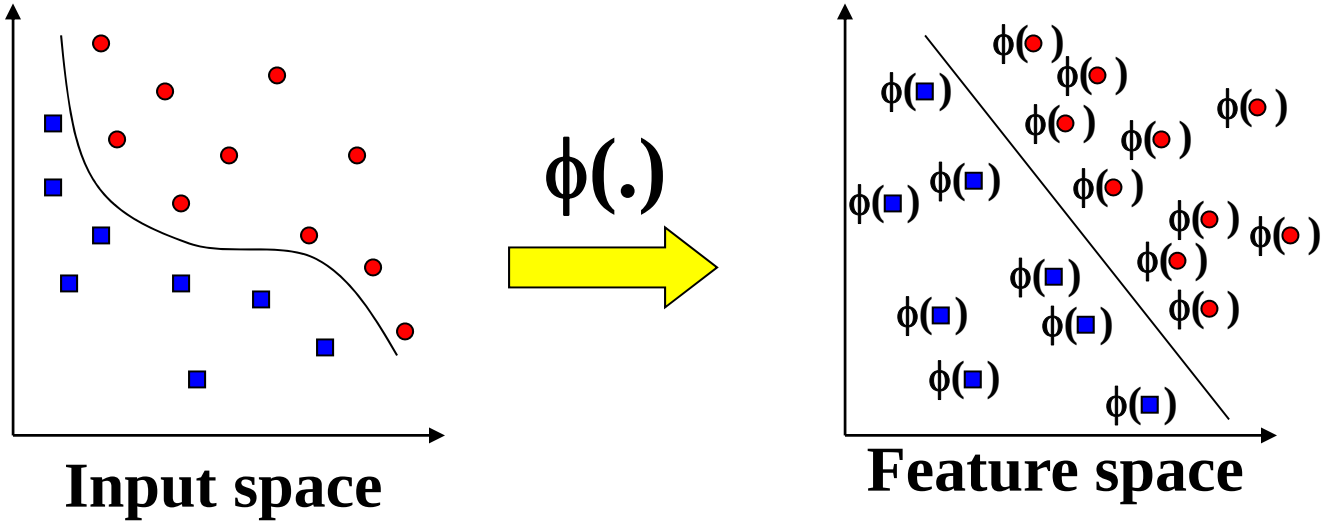
a non-linear separation boundary



now linearly separable!

How to find a useful and inexpensive transformation?

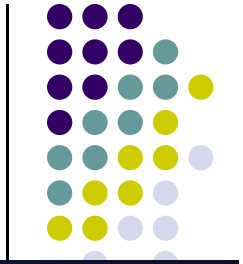
Transforming the Data



Note: feature space is of higher dimension than the input space in practice

- Computation in the feature space can be costly because it is high dimensional
  - The feature space is typically infinite-dimensional!
- The kernel trick comes to rescue

The Kernel Trick



- Recall the SVM optimization problem

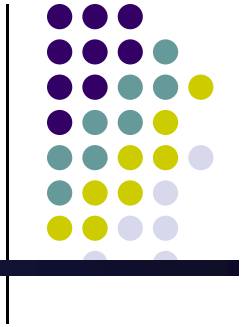
$$\max_{\alpha} \quad \mathcal{J}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

$$\text{s.t.} \quad 0 \le \alpha_i \le C, \quad i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0.$$

- The data points only appear as inner product
- As long as we can calculate the inner product in the feature space, we do not need the mapping explicitly
- Many common geometric operations (angles, distances) can be expressed by inner products
- Define the kernel function *K* by $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$

An Example for feature mapping and kernels



- Consider an input $\mathbf{x}=[x_1,x_2]$
- Suppose $\phi(.)$ is given as follows

$$\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = 1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2$$

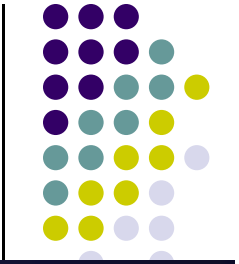
- An inner product in the feature space is

$$\left\langle \phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right), \phi\left(\begin{bmatrix} x_1' \\ x_2' \end{bmatrix}\right) \right\rangle =$$

- So, if we define the **kernel function** as follows, there is no need to carry out $\phi(.)$ explicitly

$$K(\mathbf{x},\mathbf{x}') = \left(1+\mathbf{x}^T\mathbf{x}'\right)^2$$

More examples of kernel functions



- Linear kernel (we've seen it)

$$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$$

- Polynomial kernel (we just saw an example)

$$K(\mathbf{x}, \mathbf{x}') = \left(1 + \mathbf{x}^T \mathbf{x}'\right)^p$$

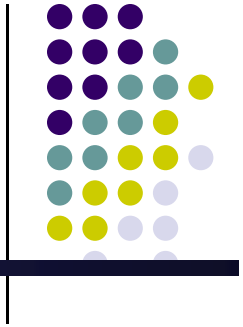
where p = 2, 3, … To get the feature vectors we concatenate all pth order polynomial terms of the components of x (weighted appropriately)

- Radial basis kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}\left\|\mathbf{x} - \mathbf{x}'\right\|^2\right)$$

In this case the feature space consists of functions and results in a non-parametric classifier.

The essence of kernel



- Feature mapping, but "without paying a cost"
  - E.g., polynomial kernel

$$K(x,z) = (x^T z + c)^d$$

  - How many dimensions we've got in the new space?
  - How many operations it takes to compute K()?

- Kernel design, any principle?
  - K(x,z) can be thought of as a similarity function between x and z
  - This intuition can be well reflected in the following "Gaussian" function

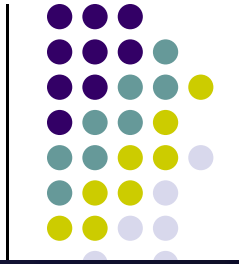
    (Similarly one can easily come up with other K() in the same spirit)

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

  - Is this necessarily lead to a "legal" kernel?

    (in the above particular case, K() is a legal one, do you know how many dimension ϕ(x) is?

Kernel matrix

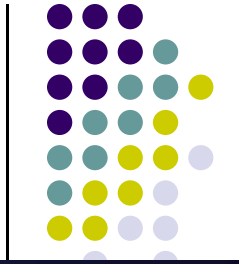


- Suppose for now that K is indeed a valid kernel corresponding to some feature mapping ϕ, then for $x_1, \ldots, x_m$, we can compute an $m \times m$ matrix $K = \{K_{i,j}\}$, where $K_{i,j} = \phi(x_i)^T \phi(x_j)$
- This is called a kernel matrix!
- Now, if a kernel function is indeed a valid kernel, and its elements are dot-product in the transformed feature space, it must satisfy:
  - Symmetry $K = K^T$

    proof $K_{i,j} = \phi(x_i)^T \phi(x_j) = \phi(x_j)^T \phi(x_i) = K_{j,i}$
  - Positive –semidefinite $y^T K y \geq 0 \quad \forall y$

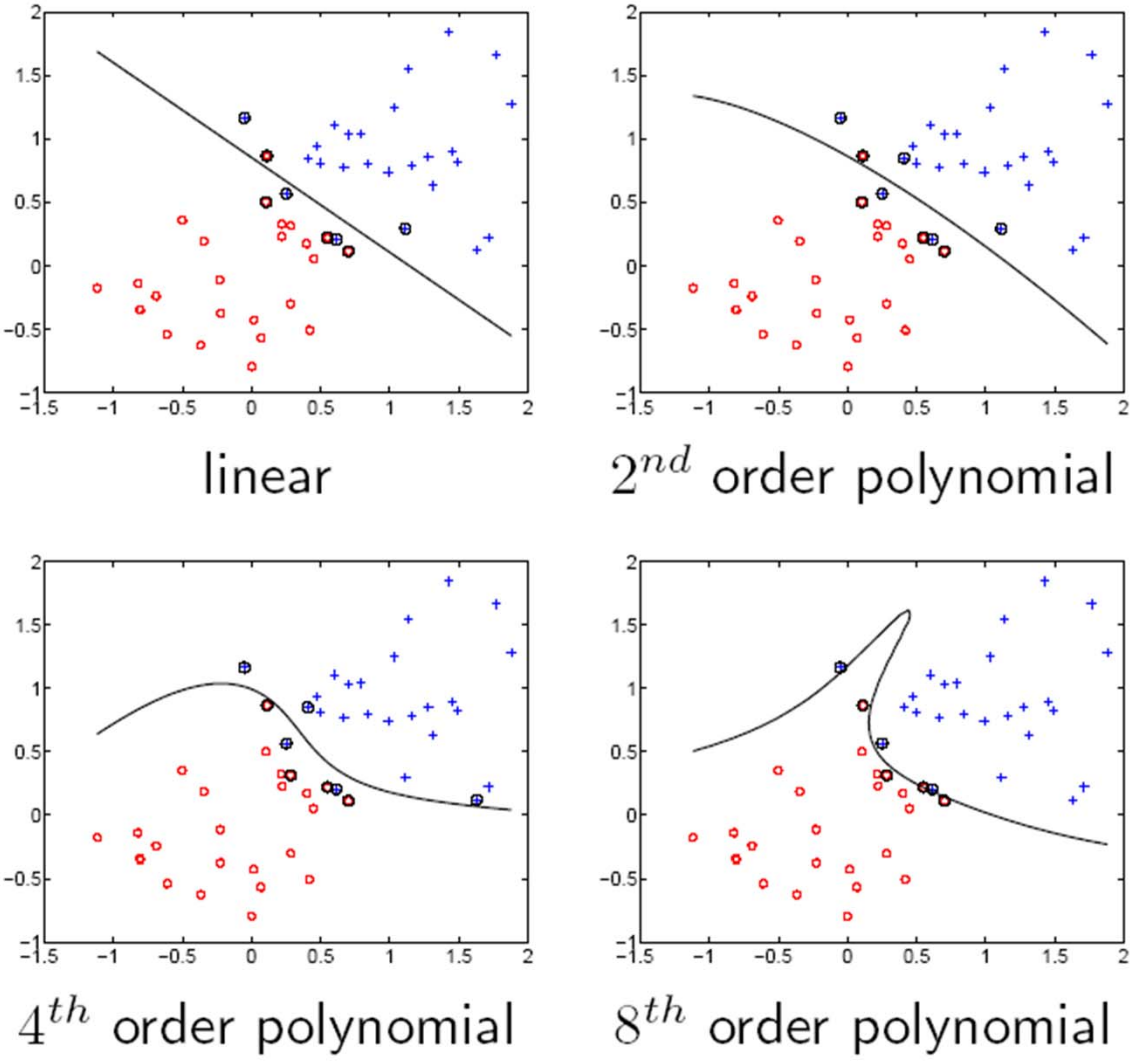
    proof?

Mercer kernel

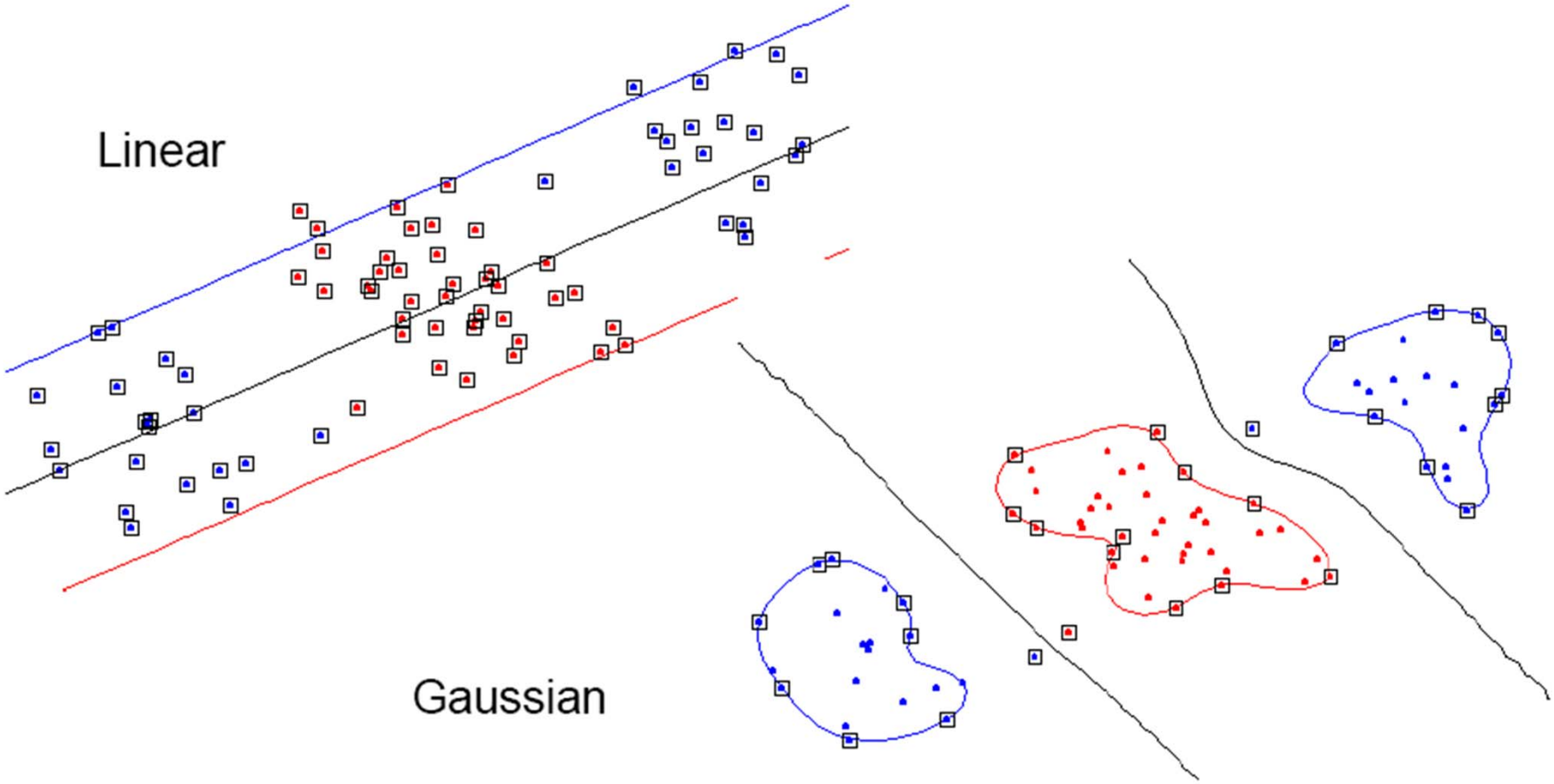


Theorem (Mercer): Let K: $\mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^n$ be given. Then for K to be a valid (Mercer) kernel, it is necessary and sufficient that for any $\{x_i, \ldots, x_m\}$, $(m < \infty)$, the corresponding kernel matrix is symmetric positive semi-denite.

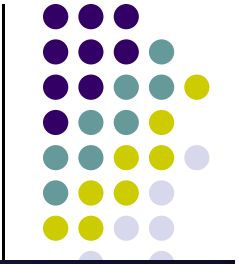
SVM examples



Examples for Non Linear SVMs – Gaussian Kernel



Remember the Kernel Trick!!!



Primal Formulation:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C\sum_j \xi$$

$$(\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_j) + b)y_j \geqslant 1 - \xi_j \quad \forall j$$

$$\xi_j \geqslant 0 \quad \forall j$$

Infinite, cannot be directly computed

But the dot product is easy to compute ☺

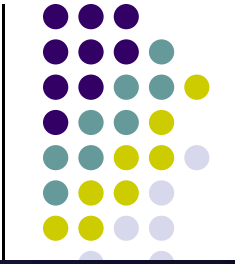
Dual Formulation:

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_i)$$

$$\sum_i \alpha_i y_i = 0$$

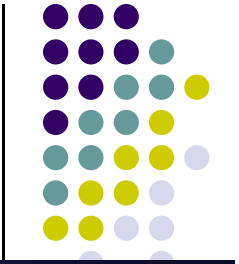
$$0 \leqslant \alpha_i \leqslant C \quad \forall i$$

Overview of Hilbert Space Embedding



- Create an infinite dimensional statistic for a distribution.
- Two Requirements:
  - Map from distributions to statistics is one-to-one
  - Although statistic is infinite, it is cleverly constructed such that the kernel trick can be applied.
- Perform Belief Propagation as if these statistics are the conditional probability tables.
- We will now make this construction more formal by introducing the concept of Hilbert Spaces

Vector Space



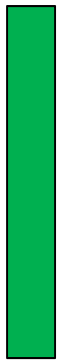
- A set of objects closed under linear combinations (e.g., addition and scalar multiplication):

$$\boldsymbol{v}, \boldsymbol{w} \in \mathcal{V} \implies \alpha \boldsymbol{v} + \beta \boldsymbol{w} \in \mathcal{V}$$

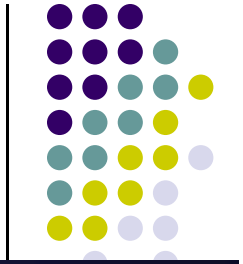
- Obeys distributive and associative laws,

- Normally, you think of these "objects" as finite dimensional vectors. However, in general the objects can be functions.
  - **Nonrigorous Intuition**: A function is like an infinite dimensional vector.

$$\boldsymbol{f} =$$

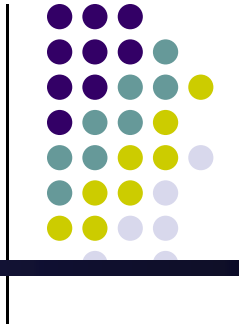


Hilbert Space



- A Hilbert Space is a complete vector space equipped with an inner product.
- The inner product $\langle \boldsymbol{f}, \boldsymbol{g} \rangle$ has the following properties:
  - Symmetry $\langle \boldsymbol{f}, \boldsymbol{g} \rangle = \langle \boldsymbol{g}, \boldsymbol{f} \rangle$
  - Linearity $\langle \alpha \boldsymbol{f}_1 + \beta \boldsymbol{f}_2, \boldsymbol{g} \rangle = \alpha \langle \boldsymbol{f}_1, \boldsymbol{g} \rangle + \beta \langle \boldsymbol{f}_2, \boldsymbol{g} \rangle$
  - Nonnegativity $\langle \boldsymbol{f}, \boldsymbol{f} \rangle \geqslant 0$
  - Zero $\langle \boldsymbol{f}, \boldsymbol{f} \rangle = 0 \implies \boldsymbol{f} = 0$
- Basically a “nice” infinite dimensional vector space, where lots of things behave like the finite case
  - e.g. using inner product we can define “norm” or “orthogonality”
  - e.g. a norm can be defined, allows one to define notions of convergence

Hilbert Space Inner Product



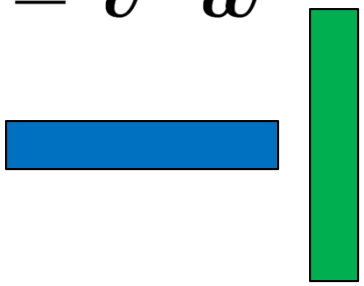
- Example of an inner product (just an example, inner product not required to be an integral)

$$\langle \boldsymbol{f}, \boldsymbol{g} \rangle = \int \boldsymbol{f}(x)\boldsymbol{g}(x)\, dx$$

Inner product of two functions is a number

- Traditional finite vector space inner product

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle = \boldsymbol{v}^\top \boldsymbol{w}$$



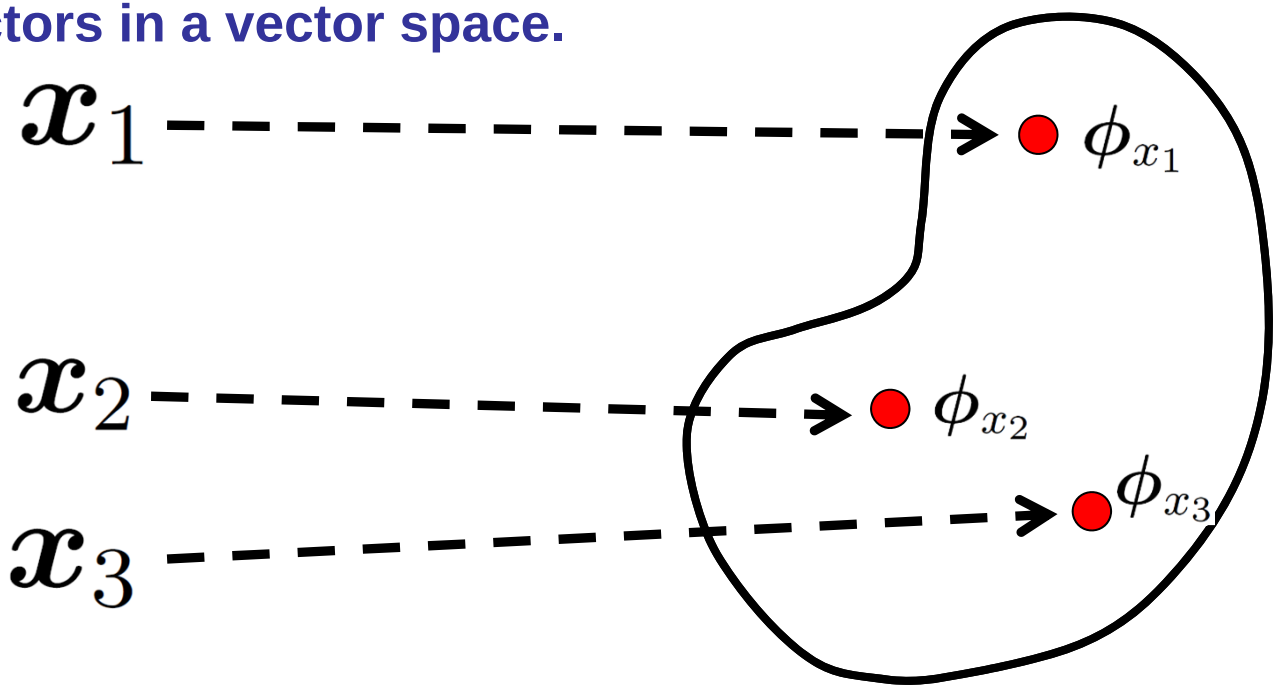
= **scalar**

Recall the SVM kernel Intuition

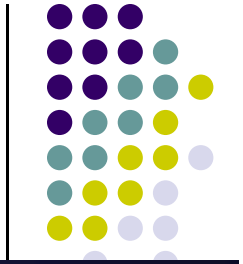
$$\min_{\boldsymbol{w},b} \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C\sum_j \xi$$

$$(\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_j) + b)y_j \geqslant 1 - \xi_j \quad \forall j \qquad \xi_j \geqslant 0 \quad \forall j$$

Maps data points to Feature Functions, which corresponds to some vectors in a vector space.



The Feature Function



- Consider holding one element of the kernel fixed. We get a function of one variable which we call the feature function. The collection of feature functions is called the **feature map**.

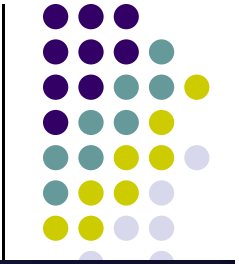
$$\phi_x := \boldsymbol{K}(x, \cdot)$$

- For a Gaussian Kernel the feature functions are unnormalized Gaussians:

$$\phi_1(y) = \exp\left(\frac{\|1 - y\|_2^2}{\sigma^2}\right)$$

$$\phi_{1.5}(y) = \exp\left(\frac{\|1.5 - y\|_2^2}{\sigma^2}\right)$$

Reproducing Kernel Hilbert Space



- Given a kenel $k(x,x')$, we now construct a Hilbert space such that k defines an inner product in that space
  - We begin with a kernel map:

$$\Phi \ : \ x \rightarrow k(\cdot, x)$$

  - We now construct a vector space containing all linear combinations of the functions $k(\ ,x)$:

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i)$$

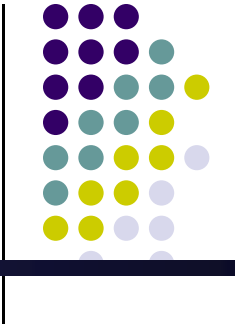
  - We now **define** an inner product. Let $g(\cdot) = \sum_{j=1}^{m'} \beta_j k(\cdot, x'_j)$ we have

$$\langle f, g \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j)$$

    please verify this in fact is an inner product: satisfying symmetry, linearity, and zero-norm law : $\langle f, f \rangle = 0 \ \Rightarrow f = 0$

    (here we need "reproducing property", and Cauchy-Schwartz inequaliy

Reproducing Kernel Hilbert Space



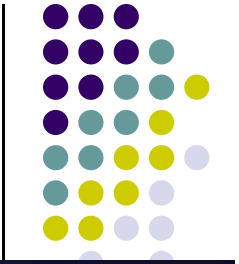
- The $k(\ ,x)$ is a **reproducing** kernel map:

$$\langle k(\cdot, x), f \rangle = \sum_{i=1}^{m} \alpha_i k(x_i) = f(x)$$

  - This shows that the kernel is a *representer of evaluation (or, evaluation function)*
  - This is analogous to the Dirac delta function.
  - If we plug in the kernel in for f: $\langle k(\cdot, x), k(\cdot, x') \rangle = k(x, x')$

- With such a definition of inner product, we have constructed a subspace of the Hilbert space --- a **reproducing kernel Hilbert space** (RKHS)

Back to Feature Map



- The collection of evaluation functions is the feature map!!!

$$\min_{\boldsymbol{w},b} \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C\sum_j \xi$$

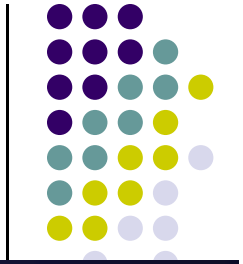
$$(\boldsymbol{w}^\top \phi(\boldsymbol{x}_j) + b)y_j \geqslant 1 - \xi_j \quad \forall j$$

$$\xi_j \geqslant 0 \quad \forall j$$

The Feature Map is the collection of Evaluation Functions!

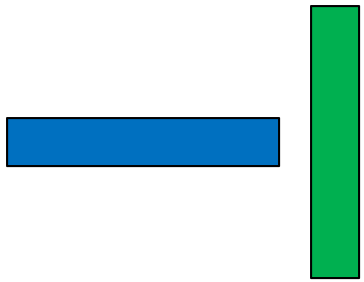
- **Intuition:** A more complicated feature map/kernel corresponds to ``richer" RKHS
- **Basically, a "really nice" infinite dimensional vector space where even more things behave like the finite case**

Inner Product of Feature Maps



- Define the Inner Product as:

$$\langle \phi_x, \phi_y \rangle = \langle \boldsymbol{K}(x,\cdot), \boldsymbol{K}(y,\cdot) \rangle := \boldsymbol{K}(x,y)$$

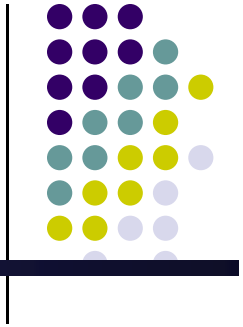


$=$ **scalar**

- Note that:

$$\phi_x(y) = \phi_y(x) = \boldsymbol{K}(x,y)$$

Mercer's theorem and RKHS



- Recall the following condition for Mercer's theorem for K

$$\int \int \boldsymbol{K}(x,y)\boldsymbol{f}(x)\boldsymbol{f}(y)\,dx\,dy > 0 \quad \forall \boldsymbol{f}$$

- We can also "construct" our Reproducing Kernel Hilbert Space with a **Mercer Kernel**, as a linear combination of its eigen-functions:

$$\int k(x,x')\phi_i(x') = \sum_{j=1}^{\infty} \lambda\phi_j(x)$$

which can be shown to entail reproducing property (homework?)

Summary: RKHS

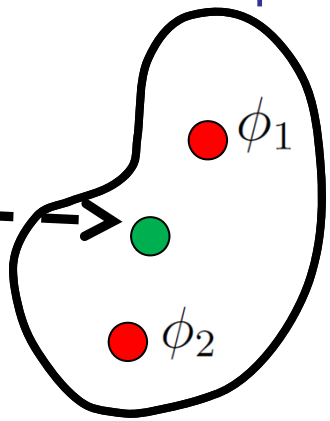
- Consider the set of functions that can be formed with linear combinations of these feature functions:

$$\mathcal{F}_0 = \left\{ f(z) : \sum_{j=1}^{k} \alpha_j \phi_{x_j}(z), \forall k \in \mathbb{N}_+ \text{ and } x_j \in \mathcal{X} \right\}$$

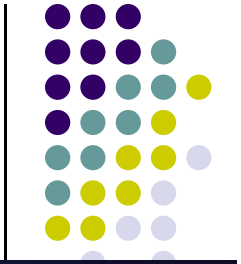
- We define the Reproducing Kernel Hilbert Space $\mathcal{F}$ to the completion of $\mathcal{F}_0$ (like $\mathcal{F}_0$ with the "holes" filled in)

- Intuitively, the feature functions are like an over-complete basis for the RKHS

$$\boldsymbol{f}(z) = \alpha_1 \phi_1(z) + \alpha_2 \phi_2(z)$$



Summary: Reproducing Property

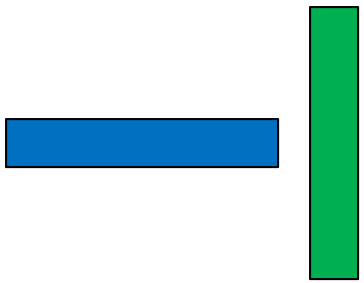


- It can now be derived that the inner product of a function **f** with $\boldsymbol{\phi}_X$, evaluates a function at point ***x***:

$$\begin{aligned}
\langle \boldsymbol{f}, \boldsymbol{\phi}_x \rangle &= \left\langle \sum_j \alpha_j \boldsymbol{\phi}_{x_j}, \boldsymbol{\phi}_x \right\rangle \\
&= \sum_j \alpha_j \langle \boldsymbol{\phi}_{x_j}, \boldsymbol{\phi}_x \rangle \qquad \textbf{Linearity of inner product} \\
&= \sum_j \alpha_j \boldsymbol{K}(x_j, x) \qquad \textbf{Definition of kernel} \\
&= \boldsymbol{f}(x)
\end{aligned}$$

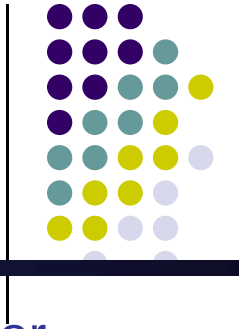
Remember that

$$\boldsymbol{K}(x_j, x) := \boldsymbol{\phi}_{x_j}(x)$$



= **scalar**

Summary: Evaluation Function



- A Reproducing Kernel Hilbert Space is an Hilbert Space where for any ***X***, the evaluation functional indexed by ***X*** takes the following form:

$$\text{Eval}_X(\cdot) = \langle \phi_X, \rangle$$

Evaluation Function, must be a function in the RKHS

Same evaluation function for different functions (but same point)

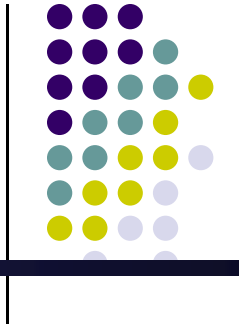
$$\boldsymbol{f}(X_1) = \langle \phi_{X_1}, \boldsymbol{f} \rangle$$
$$\boldsymbol{g}(X_1) = \langle \phi_{X_1}, \boldsymbol{g} \rangle$$

Different points are associated with different evaluation functions

$$\boldsymbol{f}(X_2) = \langle \phi_{X_2}, \boldsymbol{f} \rangle$$
$$\boldsymbol{g}(X_2) = \langle \phi_{X_2}, \boldsymbol{g} \rangle$$

- Equivalent (More Technical) Definition: An RKHS is a Hilbert Space where the evaluation functionals are bounded. (The previous definition then follows from Riesz Representation Theorem)

RKHS or Not?



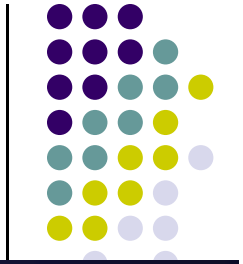
- Is the vector space of 3 dimensional real valued vectors an RKHS?

Yes!!!

$$\text{Eval}_i(\cdot) = \langle \boldsymbol{e}_i, \cdot \rangle$$

Homework !

RKHS or Not?



- Is the space of functions such that

$$\int |f(z)|^2 \, dz < \infty$$

an RKHS?

No!!!! **Homework !**

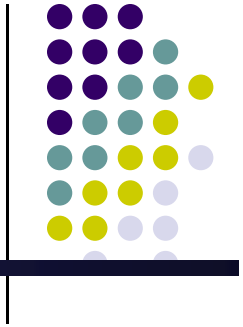
But, can't the evaluation functional be an inner product with the delta function?

$$\text{Eval}_X(\cdot) = \langle \delta_X, \cdot \rangle$$

$$f(X) = \int f(z)\delta_X(z) \, dz$$

The problem is that the delta function is not in my space!

The Kernel



- I can evaluate my evaluation function with another evaluation function!

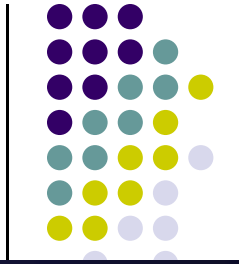
$$k(X_1, X_2) := \phi_{X_1}(X_2) = \phi_{X_2}(X_1) = \langle \phi_{X_1}, \phi_{X_2} \rangle = \int \phi_{X_1}(z)\phi_{X_2}(z)\, dz$$

- Doing this for all pairs in my dataset gives me the Kernel Matrix **K**:

$$\boldsymbol{K} = \begin{pmatrix} k(X_1, X_1) & k(X_1, X_2) & k(X_1, X_3) \\ k(X_1, X_2) & k(X_1, X_2) & k(X_1, X_3) \\ k(X_1, X_1) & k(X_1, X_2) & k(X_1, X_3) \end{pmatrix}$$

- There may be infinitely many evaluation functions, but I only have a finite number of training points, so the kernel matrix is finite!!!!

Correspondence between Kernels and RKHS

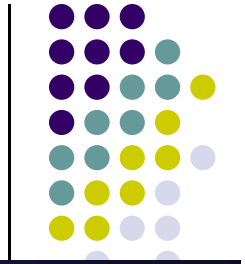


- A kernel is positive semi-definite if the kernel matrix is positive semidefinite for any choice of finite set of observations.

- **Theorem (Moore-Aronszajn):** Every positive semi-definite kernel corresponds to a unique RKHS, and every RKHS is associated with a unique positive semi-definite kernel.

- Note that the kernel does not uniquely define the feature map (but we don't really care since we never directly evaluate the feature map anyway).

RKHS norm and SVM



- Recall that in SVM:

$$f(\cdot) = \langle w, x \rangle = \sum_{i=1}^{m} \alpha_i y_i k(\cdot, x_i)$$

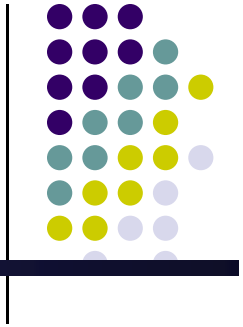
Therefore $f(\cdot) \in \mathcal{H}$

Moreover:

$$\|f(\cdot)\|_{\mathcal{H}}^2 = \langle \sum_{i=1}^{m} \alpha_i y_i k(\cdot, x_i), \sum_{j=1}^{m} \alpha_j y_j k(\cdot, x_j) \rangle$$

$$=$$

Primal and dual SVM objective

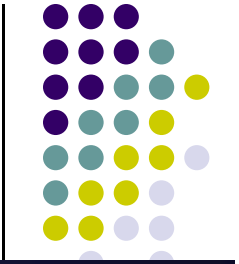


- In our primal problem, we minimize $w^T w$ subject to constraints. This is equivalent to:

$$\begin{aligned} \|w\|^2 &= w^T w = \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \Phi(x_i)\Phi(x_j)\rangle \\ &= \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ &= \|f\|_{\mathcal{H}}^2 \end{aligned}$$

which is equivalent to minimizing the Hilbert norm of f subject to constraints

The Representer Theorem



- In the general case, for a primal problem P of the form:

$$\min_{f \in \mathcal{H}} \{C(f, \{x_i, y_i\}) + \Omega(\|f\|_{\mathcal{H}})\}$$

where $\{x_i, y_i\})_{i=1}^m$ are the training data.

If the following conditions are satisfied:

- The loss function C is point-wise, i.e., $C(f, \{x_i, y_i\}) = C(\{x_i, y_i, f(x_i)\})$
- $\Omega(\cdot)$ is monotonically increasing

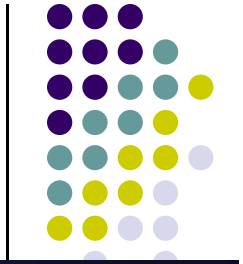
- The representer theorem (Kimeldorf and Wahba, 1971):

every minimizer of P admits a representation of the form

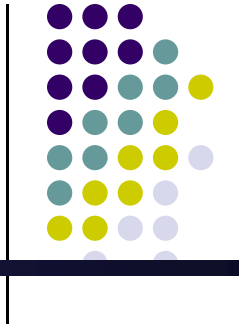
$$f(\cdot) = \sum_{i=1}^{m} \alpha_i K(\cdot, x_i)$$

i.e., a linear combination of (a finite set of) function given by the data

Proof of Representer Theorem

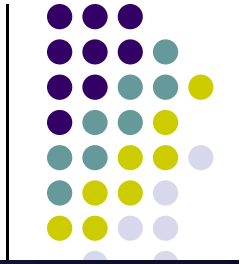


Another view of SVM



- Q: why SVM is "dual-sparse", i.e., having a few support vectors (most of the α's are zero).
  - The SVM loss $w^T w$ does not seem to imply that
  - And the representer theorem does not either!

Another view of SVM: L_1 regularization



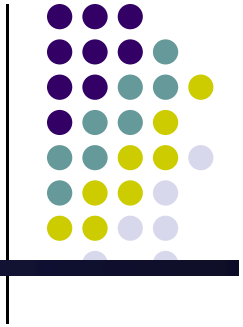
- The basis-pursuit denoising cost function (chen & Donoho):

$$J(\alpha) = \frac{1}{2}\|f(\cdot) - \sum_{i=1}^{N} \alpha_i \phi_i()\|_{L_2}^2 + \lambda \|\alpha\|_{L_1}$$

- Instead we consider the following modified cost:

$$J(\alpha) = \frac{1}{2}\sum \|f(\cdot) - \sum_{i=1}^{N} \alpha_i K(\cdot, x_i)\|_{\mathcal{H}}^2 + \lambda \|\alpha\|_{L_1}$$

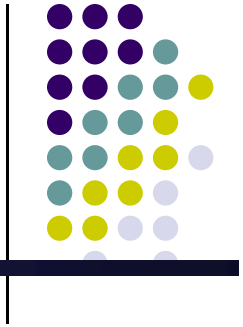
RKHS norm interpretation of SVM



$$J(\alpha) = \frac{1}{2}\sum \|f(\cdot) - \sum_{i=1}^{N} \alpha_i K(\cdot, x_i)\|_{\mathcal{H}}^2 + \lambda \|\alpha\|_{L_1}$$

- The RKHS norm of the first term can now be computed exactly!

RKHS norm interpretation of SVM

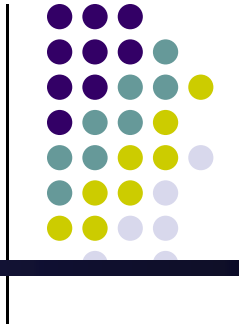


- Now we have the following optimization problem:

$$\min_{\alpha} \left\{ -\sum_{i} \alpha_i y_i + \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) + \sum_{i} \lambda |\alpha_i| \right\}$$

This is exactly the dual problem of SVM!

Take home message



- Kernel is a (nonlinear) feature map into a Hilbert space
- Mercer kernels are “legal”
- RKHS is a Hilbert equipped with an “inner product” operator defined by mercer kernel
- Reproducing property make kernel works like an evaluation function
- Representer theorem ensures optimal solution to a general class of loss function to be in the Hilbert space
- SVM can be recast as an L1-regularized minimization problem in the RKHS