Introduction to kernel learning.

Oriol Pujol

Introduction to Machine Learning

2013

Acknowledgements

This work is inspired in the courses of T. Jaakkola, M. Collins, L. Kaelbling, and T. Poggio at MIT, Andrew Ng at Stanford, Y. Abu-Mostafa at CalTech, E. Xing at CMU, and all my mentors and people who made me realize Machine Learning is one of my passions.

Outline

- Basics of functional analysis. Reproducing Kernel Hilbert Spaces
- RBF additive models: Kernel Regularized Least Squares
- A view to the kernel matrix
- The kernel trick
- Kernel perceptron
- Kernel Support Vector Machines

Regularization

Data: $(\mathbf{x}_1, y_1) \dots (\mathbf{x}_n, y_n) \in \mathbb{R}^N \times \mathbb{R}$
Estimate: $f : \mathbb{R}^N \to \mathbb{R}$

Hypothesis space $\mathcal{H}$.

$$f^* = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_i \mathcal{L}(f(x_i), y_i) + \Omega(f)$$

fit to data + complexity penalty

Occam's razor: "simpler" hypotheses are better.
$\Omega(f)$ incorporates our simplicity assumptions.
If we fail with the complexity penalty we still have λ (fixed by validation)

Regularization

Complex functions "wiggle/oscillate" more.

How do we measure the complexity/oscillation of a function?

Regularization

Complex functions "wiggle/oscillate" more.

How do we measure the complexity/oscillation of a function?

Enter the Hilbert space!

Basics on Functional Analysis (part I)

A comprehensive review

- Functional analysis is the branch of mathematics concerned with the study of spaces of functions.
- In a nutshell, functions are seen as points in a vector space (usually of infinite dimension) on a domain $\mathcal{D}$.

Basics of Functional Analysis (part II)

A comprehensive review

Examples of function spaces:

- $\mathcal{R}$ – The space of functions from $\mathbb{R}$ to $\mathbb{R}$.
- $C^n(\mathcal{D})$ – The space of n times differentiable functions on $\mathcal{D}$.
- $L^1(\mathcal{D})$ – The space of all absolutely integrable functions on $\mathcal{D}$. (i.e. $\{f | \int_{\mathcal{D}} |f(x)| dx < \infty)$
- $L^2(\mathcal{D})$ – The space of all square integrable functions on $\mathcal{D}$. (i.e. $\{f | \int_{\mathcal{D}} |f(x)|^2 dx < \infty)$

All these spaces form a vector space over $\mathbb{R}$. Let $f, g \in \mathcal{R}$. The basic axioms of the vector space are the well known:

- $(f + g)(x) = f(x) + g(x), \forall x \in \mathbb{R}$
- $(\alpha f)(x) = \alpha f(x), \forall x \in \mathbb{R}$

Basics of Functional Analysis (part II)

A comprehensive review

Examples of function spaces:

- $\mathcal{R}$ – The space of functions from $\mathbb{R}$ to $\mathbb{R}$.
- $C^n(\mathcal{D})$ – The space of n times differentiable functions on $\mathcal{D}$.
- $L^1(\mathcal{D})$ – The space of all absolutely integrable functions on $\mathcal{D}$. (i.e. $\{f | \int_{\mathcal{D}} |f(x)| dx < \infty)$
- $L^2(\mathcal{D})$ – The space of all square integrable functions on $\mathcal{D}$. (i.e. $\{f | \int_{\mathcal{D}} |f(x)|^2 dx < \infty)$

All these spaces form a vector space over $\mathbb{R}$. Let $f, g \in \mathcal{R}$. The basic axioms of the vector space are the well known:

- $(f + g)(x) = f(x) + g(x), \forall x \in \mathbb{R}$
- $(\alpha f)(x) = \alpha f(x), \forall x \in \mathbb{R}$

Basics of Functional Analysis (part III)

A comprehensive review

"Not very serious" definition of functional

A functional is a function that takes one or more functions as arguments, and which returns a scalar value.

Concepts of linear algebra such as *norm* and *inner product* can be defined for functions in function spaces.

Normed vector space.

A normed vector space is a pair $(V, \|\cdot\|)$ where V is a vector space and $\|\cdot\|$ is the associated norm, satisfying the following properties for all $u, v \in V$:

1. $\|v\| \geqslant 0$
2. $\|u + v\| \leqslant \|u\| + \|v\|$ (triangle inequality)
3. $\|\|\alpha|v\| = |\alpha|\|v\|$ (positive homogeneity)
4. $\|v\| = 0 \iff v = 0$

Basics of Functional Analysis (part III)

A comprehensive review

"Not very serious" definition of functional

A functional is a function that takes one or more functions as arguments, and which returns a scalar value.

Concepts of linear algebra such as *norm* and *inner product* can be defined for functions in function spaces.

Normed vector space.

A normed vector space is a pair $(V, \|\cdot\|)$ where V is a vector space and $\|\cdot\|$ is the associated norm, satisfying the following properties for all $u, v \in V$:

1. $\|v\| \geqslant 0$
2. $\|u + v\| \leqslant \|u\| + \|v\|$ (triangle inequality)
3. $\|\|\alpha|v\| = |\alpha|\|v\|$ (positive homogeneity)
4. $\|v\| = 0 \iff v = 0$

Basics of Functional Analysis (part III)

A comprehensive review

"Not very serious" definition of functional

A functional is a function that takes one or more functions as arguments, and which returns a scalar value.

Concepts of linear algebra such as *norm* and *inner product* can be defined for functions in function spaces.

Normed vector space.

A normed vector space is a pair $(V, \|\cdot\|)$ where V is a vector space and $\|\cdot\|$ is the associated norm, satisfying the following properties for all $u, v \in V$:

1. $\|v\| \geqslant 0$
2. $\|u + v\| \leqslant \|u\| + \|v\|$ (triangle inequality)
3. $\|\|\alpha|v\| = |\alpha|\|v\|$ (positive homogeneity)
4. $\|v\| = 0 \iff v = 0$

Basics of Functional Analysis (part IV)

A comprehensive review

Inner product space.

An real inner product space is a pair $(V, \langle \cdot, \cdot \rangle)$, where V is a real vector space and $\langle \cdot, \cdot \rangle$ the associated inner product, satisfying the following properties for all $u, v, \in V$:

1. $\langle u, v \rangle = \langle v, u \rangle$ (symmetry)
2. $\langle \alpha u, v \rangle = \alpha \langle u, v \rangle, \quad \langle u, \alpha v \rangle = \alpha \langle u, v \rangle$ (bilinearity)
   $\langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle, \quad \langle u, v + w \rangle = \langle u, w \rangle + \langle v, w \rangle$
3. $\langle u, u \rangle \geqslant 0 \quad$ and $\quad \langle u, u \rangle = 0 \iff u = 0$

The inner product introduces a norm by setting $\|v\| = \langle v, v \rangle^{1/2}$.
Furthermore, the Cauchy-Schwarz inequality holds in real inner product spaces.

$$|\langle u, v \rangle| \leqslant \|u\| \|v\|$$

Basics of Functional Analysis (part V)

Hilbert Space

A Hilbert space is an inner product space that is complete with respect to the induced metric.

Note that a Hilbert space does not depen on functions. (i.e. $\mathbb{R}^n$ is a Hilbert space)

Reproducing Kernel Hilbert Space

A linear evaluation functional over the *Hilbert space of functions* $\mathcal{H}$ is a linear functional $\mathcal{F}_x : \mathcal{H} \rightarrow \mathbb{R}$ that evaluates each function in the space at the point x, or

$$\mathcal{F}_x[f] = f(x)$$

Informal: It is important to understand the difference between f and $f(x)$. While f represents the abstraction of a function, $f(x)$ corresponds to the evaluation of a function on a point x

Definition

A Hilbert space $\mathcal{H}$ is a *reproducing kernel Hilbert space* (RKHS) if the evaluation functionals are bounded, i.e. if there exists a M such that

$$|\mathcal{F}_x[f]| = |f(x)| \leqslant M\|f\|_{\mathcal{H}} \forall f \in \mathcal{H}$$

Reproducing Kernel Hilbert Space

A linear evaluation functional over the *Hilbert space of functions* $\mathcal{H}$ is a linear functional $\mathcal{F}_x : \mathcal{H} \rightarrow \mathbb{R}$ that evaluates each function in the space at the point x, or

$$\mathcal{F}_x[f] = f(x)$$

Informal: It is important to understand the difference between f and $f(x)$. While f represents the abstraction of a function, $f(x)$ corresponds to the evaluation of a function on a point x

Definition

A Hilbert space $\mathcal{H}$ is a *reproducing kernel Hilbert space* (RKHS) if the evaluation functionals are bounded, i.e. if there exists a M such that

$$|\mathcal{F}_x[f]| = |f(x)| \leqslant M\|f\|_{\mathcal{H}} \forall f \in \mathcal{H}$$

Reproducing Kernel Hilbert Space

Property

If $\mathcal{H}$ is a RKHS, then for each $x \in X$ there exists, by the Riesz representation theorem a function K_x of $\mathcal{H}$ (called representer) with the reproducing property

$$\mathcal{F}_x[f] = f(x) = \langle K_x, f \rangle = \langle K(x, \cdot), f(\cdot) \rangle$$

Observe that the evaluation of a funtion on a given point corresponds to the inner product between the function and the a Reproducing Kernel function evaluated at that point.

Introduction to regularization

Data: $(\mathbf{x}_1, y_1) \dots (\mathbf{x}_n, y_n) \in \mathbb{R}^N \times \mathbb{R}$
Estimate: $f : \mathbb{R}^N \rightarrow \mathbb{R}$

Hypothesis space $\mathcal{H}$.

$$f^* = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_i \mathcal{L}(f(x_i), y_i) + \lambda \|f\|^2_{\mathcal{H}}$$

fit to data + complexity penalty

$\|f\|^2_{\mathcal{H}}$ measures the "complexity" of the function

Representer's theorem

We want to find f^*, that is the solution of the following minimization problem:

$$f^* = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_i \mathcal{L}(f(x_i), y_i) + \lambda \|f\|^2_{\mathcal{H}}$$

Remarkable fact (Representer's theorem): The optimal solution of the former problem is given by:

$$f^*(\cdot) = \sum_i \alpha_i K(x_i, \cdot)$$

Converts the problem of looking for a generic function in an infinite-dimensional space into that of finding the optimal $\alpha_i \in \mathbb{R}$ in a finite dimensional space.

Representer's theorem

Sketch of proof

Let $\mathcal{S} = span\{K(x_1, \cdot), \ldots, K(x_n, \cdot)\}$
(Recall that this is the space spanned/generated by all linear combinations of the elements)

Claim: $f \in \mathcal{H}, f \perp \mathcal{S}$ if and only if $f(x_i) = 0, \forall i$
Proof:
$f \perp \mathcal{S}$ if and only if f is orthogonal to each $K(x_i, \cdot)$,
i.e. $\langle f(\cdot), K(x_i, \cdot)\rangle_{\mathcal{H}} = 0$

Considering the reproducing property the proof is clear
$f(x_i) = \langle K(x_i, \cdot), f(\cdot)\rangle_{\mathcal{H}}$

Representer's theorem

Sketch of proof

We can decompose the hypotheses space $\mathcal{H} = \mathcal{S} \bigoplus \mathcal{S}^{\perp}$

$$f \in \mathcal{H} \qquad f = f_{\mathcal{S}} + f^{\perp}$$

Claim:

$$f(x_i) = f_{\mathcal{S}}(x_i)$$

This is clear because of the claim in the former slide.

Claim:

$$\|f_{\mathcal{S}}\|_{\mathcal{H}} \leqslant \|f\|_{\mathcal{H}}$$

Proof: Consider the triangle decomposition and remember that the $f^{\perp}$ and $f_{\mathcal{S}}$ are perpendicular, thus,

$$\|f_{\mathcal{S}}\|^2_{\mathcal{H}} + \|f^{\perp}\|^2_{\mathcal{H}} = \|f\|^2_{\mathcal{H}}$$

Representer's theorem

Sketch of proof

Remember: We want to find f^*, that is the solution of the following minimization problem:

$$f^* = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_i \mathcal{L}(f(x_i), y_i) + \lambda \|f\|^2_{\mathcal{H}}$$

This has two terms, let us make some deductions from the former claims: $\mathcal{L}(f^*(x_i), y_i)$ depends only on the evaluation of the function f on the data. Thus, using claim $f(x_i) = f_{\mathcal{S}}(x_i)$, then

$$\mathcal{L}(f^*(x_i), y_i) = \mathcal{L}(f^*_{\mathcal{S}}(x_i), y_i)$$

Thus, this term remains equal for $f^* \in \mathcal{S}$.

Representer's theorem

Sketch of proof

Remember: We want to find f^*, that is the solution of the following minimization problem:

$$f^* = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_i \mathcal{L}(f(x_i), y_i) + \lambda \|f\|^2_{\mathcal{H}}$$

What about the other term?
Recall the last claim, $\|f^*_{\mathcal{S}}\|^2_{\mathcal{H}} \leqslant \|f^*\|^2_{\mathcal{H}}$

Thus, f spaned by the basis in S has smaller (or equal) norm that the optimal solution.

In summary, the minimizer f^* must lie in $\mathcal{S}$

Representer's theorem

Sketch of proof

We see that $f^* \in \mathcal{S}$

Therefore

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

Important algorithmic implications!

Regularized Least Squares

Let us take one of the measures used to evaluate the fitting goodness of the function $f(x_i)$ i.e. $\mathcal{L}(f(x_i), y_i) = (f(x_i) - y_i)^2$, then the problem to solve is

$$f^* = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \frac{1}{2} \sum_i (f(x_i) - y_i)^2 + \frac{1}{2} \lambda \|f\|^2_{\mathcal{H}}$$

The solution of the problem is given by the Representer's theorem:
$f^*(x) = \sum_j \alpha_j K(x_j, x)$

Regularized Least Squares

Let us write down the value of $\|f^*\|^2_{\mathcal{H}}$,

$$\|f^*\|^2_{\mathcal{H}} = \langle f^*, f^* \rangle_{\mathcal{H}} = \langle \sum_j \alpha_j K(x_j, x), \sum_l \alpha_l K(x_l, x) \rangle_{\mathcal{H}}$$

by linearity

$$\sum_j \sum_l \alpha_l \alpha_j \langle K(x_j, x), K(x_l, x) \rangle_{\mathcal{H}}$$

using the reproducing property, $f(x) = \langle K(x, \cdot), f(\cdot) \rangle$,

$$\sum_j \sum_l \alpha_l \alpha_j K(x_j, x_l)$$

or equivalently in matrix form $(K(j, l) = K(x_j, x_l))$

$$\|f^*\|^2_{\mathcal{H}} = \alpha^T K \alpha$$

Regularized Least Squares

$$f^* = \arg\min_{f \in \mathcal{H}} \frac{1}{2} \sum_i (f(x_i) - y_i)^2 + \frac{\lambda}{2} \|f\|^2_{\mathcal{H}}$$

Replacing the first term,

$$\alpha^* = \arg\min_{\alpha_i \in \mathbb{R}} \frac{1}{2} \sum_i (\sum_j \alpha_j K(x_j, x_i) - y_i)^2 + \frac{\lambda}{2} \|f\|^2_{\mathcal{H}}$$

replacing the norm and using the matrix form

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}^N} \frac{1}{2} \|K\alpha - y\|^2 + \frac{\lambda}{2} \alpha^T K \alpha$$

Regularized Least Squares

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}^N} \frac{1}{2}(K\alpha - y)^T(K\alpha - y) + \frac{\lambda}{2}\alpha^T K\alpha$$

Now we find the extrema by setting the gradient with respect to α to 0

$$\begin{aligned} -K(K\alpha - y) + \lambda K\alpha &= 0 \\ (K + \lambda I)\alpha &= y \\ \alpha &= (K + \lambda I)^{-1} y \end{aligned}$$

Thus, the solution is just a matrix inversion $\mathcal{O}(n^3)$ and a matrix multiplication $\mathcal{O}(n^2)$ that depends on the number of samples n .

The matrix $K + \lambda I$ is symmetric positive definite, so the appropriate algorithm is Cholesky factorization (or SVD).

Kernel types

RKHS needs the definition of a positive semidefinite kernel function K:

$$\textbf{linear}: K(x_i, x_j) = x_i^T x_j$$
$$\textbf{polynomial}: K(x_i, x_j) = (x_i^T x_j + 1)^d$$
$$\textbf{gaussian}: K(x_i, x_j) = exp\Big(- \frac{\|x_i - x_j\|^2}{\sigma^2} \Big)$$

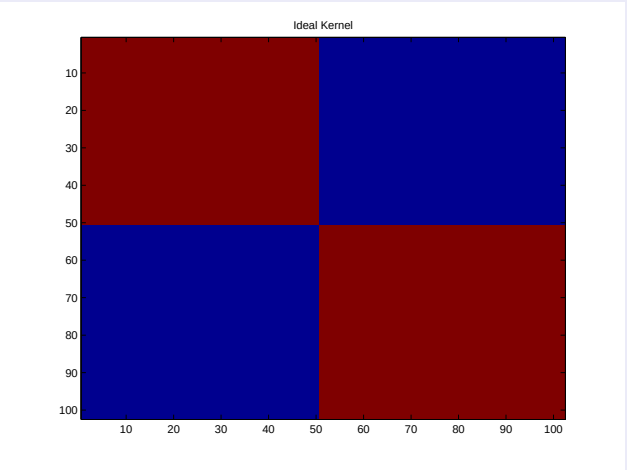
A view to the Gram matrix

Gram matrix interpretation

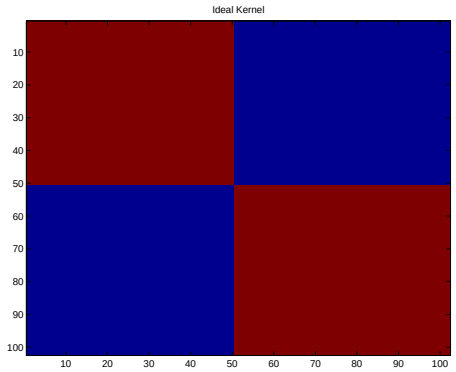
The Gram matrix shows the degree of shared information among training samples.

Gram matrix interpretation

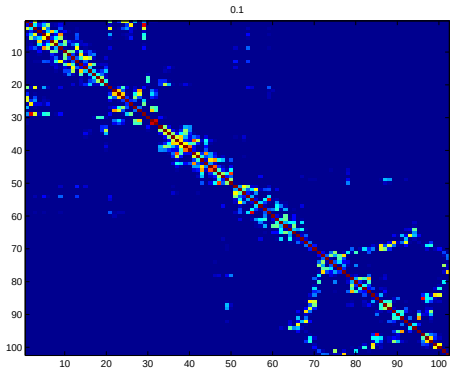
- **Bad Kernel:** Mostly diagonal $\Longrightarrow$ most points are orthogonal to each other, no clusters, no structure.
- **Good Kernel:** The matrix has structure and show clusters.
- **Ideal Kernel:** $\mathbf{K}_{\text{ideal}} = \mathbf{y}\mathbf{y}^{\mathbf{T}}$



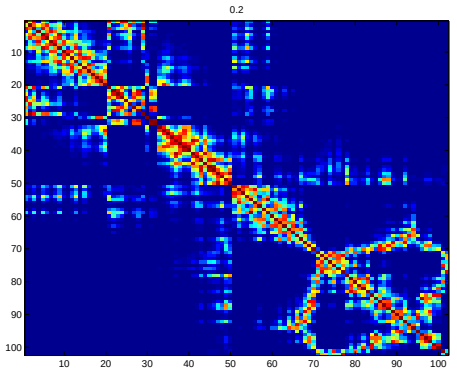
A view to the Gram matrix



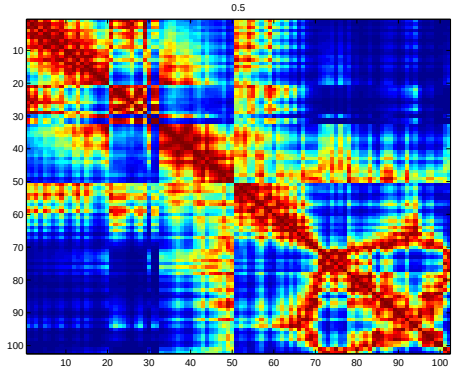
ideal



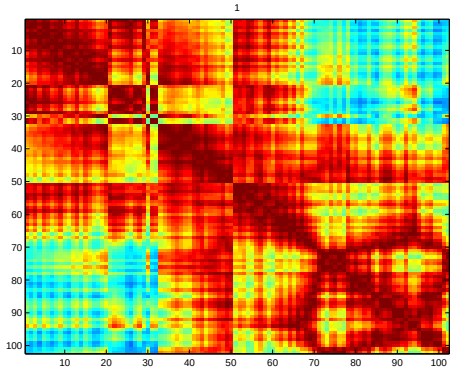
$\sigma = 0.1$



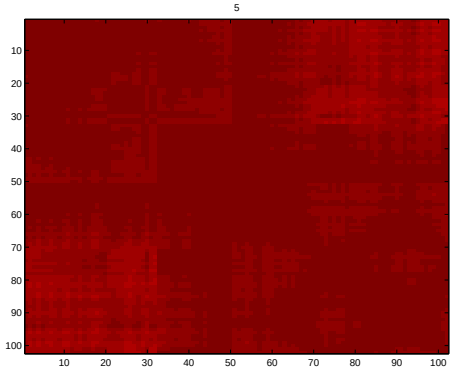
$\sigma = 0.2$



$\sigma = 0.5$



$\sigma = 1$



$\sigma = 5$

Which Gram matrix do you think is the best one?

Defining your own kernel

Designing your kernel using the properties of kernels

Let K_1 and K_2 be valid Mercer's kernels, i.e. symmetric and positive definite matrices; and $\alpha > 0$, then the following are also valid kernels

1. $K(x_i, x_j) = K_1(x_i, x_j) + K_2(x_i, x_j)$
2. $K(x_i, x_j) = K_1(x_i, x_j) \cdot K_2(x_i, x_j)$
3. $K(x_i, x_j) = \alpha K_1(x_i, x_j)$

More ideas for designing your kernels

1. Put your favorite distance $d(x_i, x_j)$ in $K(x_i, x_j) = e^{-d(x_i, x_j)}$. Done!
2. Take your favorite nonlinear transform $\Phi(x)$ and write

$$K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$$

Defining your own kernel

Designing your kernel using the properties of kernels

Let K_1 and K_2 be valid Mercer's kernels, i.e. symmetric and positive definite matrices; and $\alpha > 0$, then the following are also valid kernels

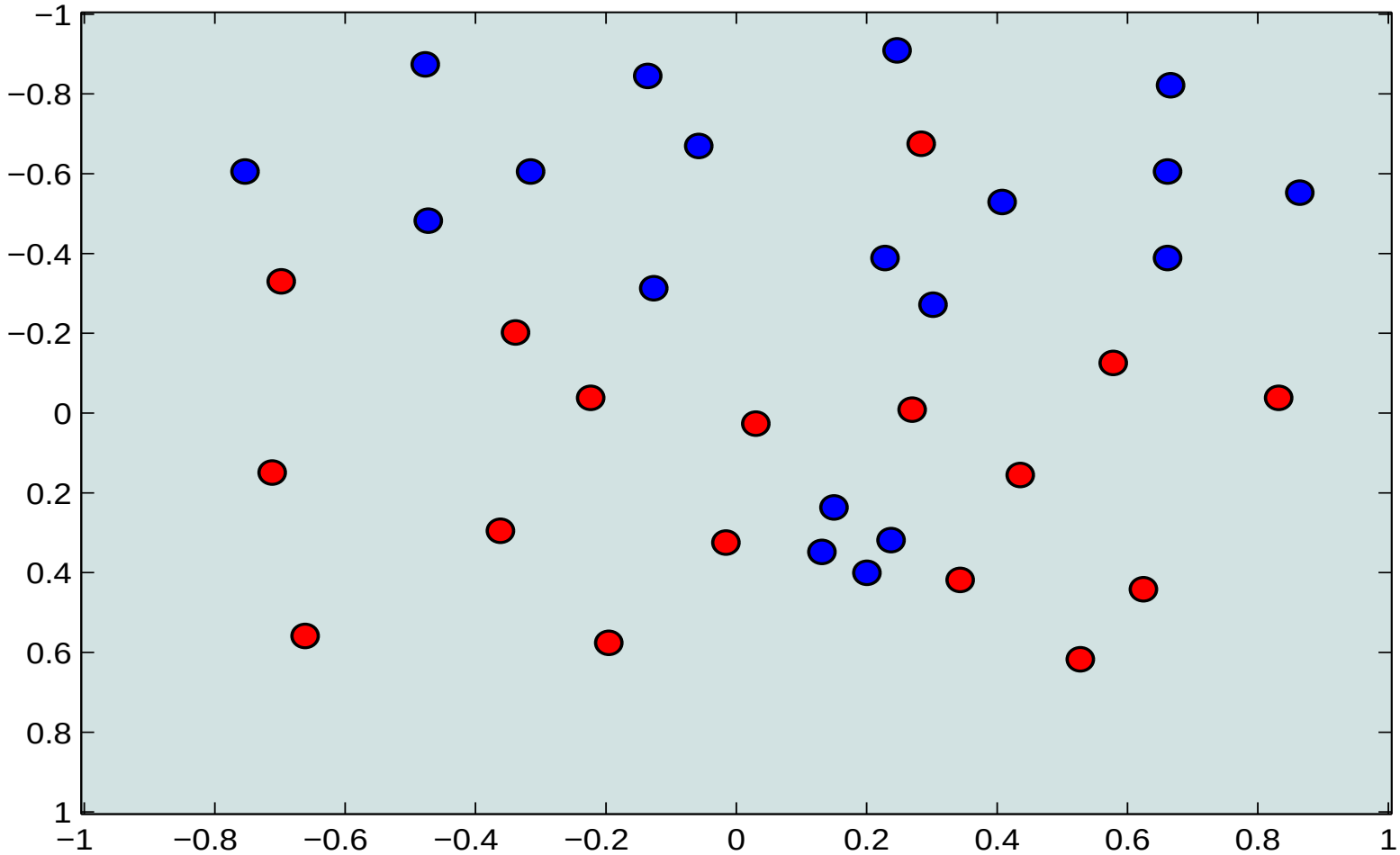
1. $K(x_i, x_j) = K_1(x_i, x_j) + K_2(x_i, x_j)$
2. $K(x_i, x_j) = K_1(x_i, x_j) \cdot K_2(x_i, x_j)$
3. $K(x_i, x_j) = \alpha K_1(x_i, x_j)$

More ideas for designing your kernels

1. Put your favorite distance $d(x_i, x_j)$ in $K(x_i, x_j) = e^{-d(x_i, x_j)}$. Done!
2. Take your favorite nonlinear transform $\Phi(x)$ and write

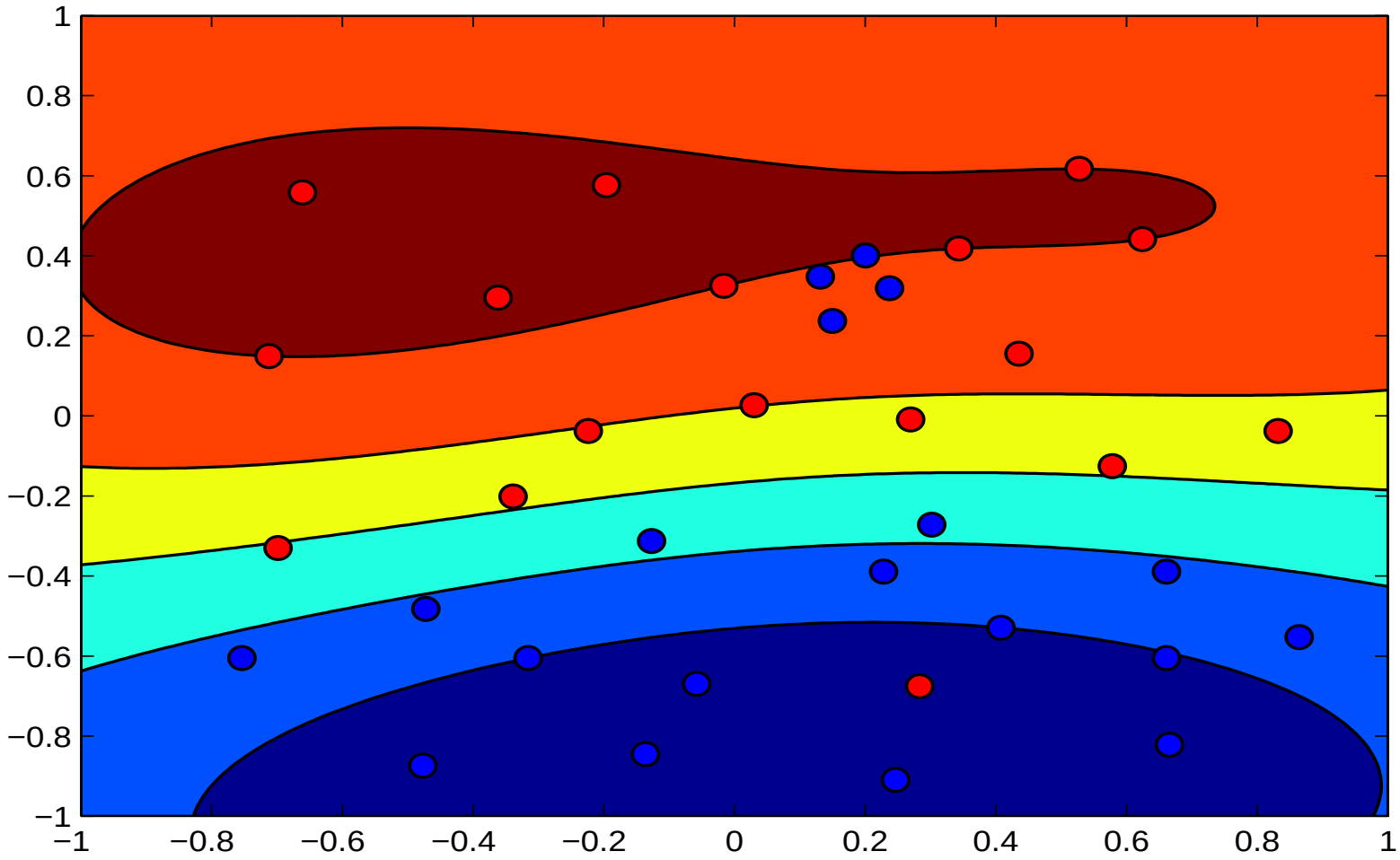
$$K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$$

Example



Example: Kernel Regularized Least Squares

Consider a Gaussian kernel (RBF) with $\sigma = 1$.
Regularization parameter $\lambda = 1$.

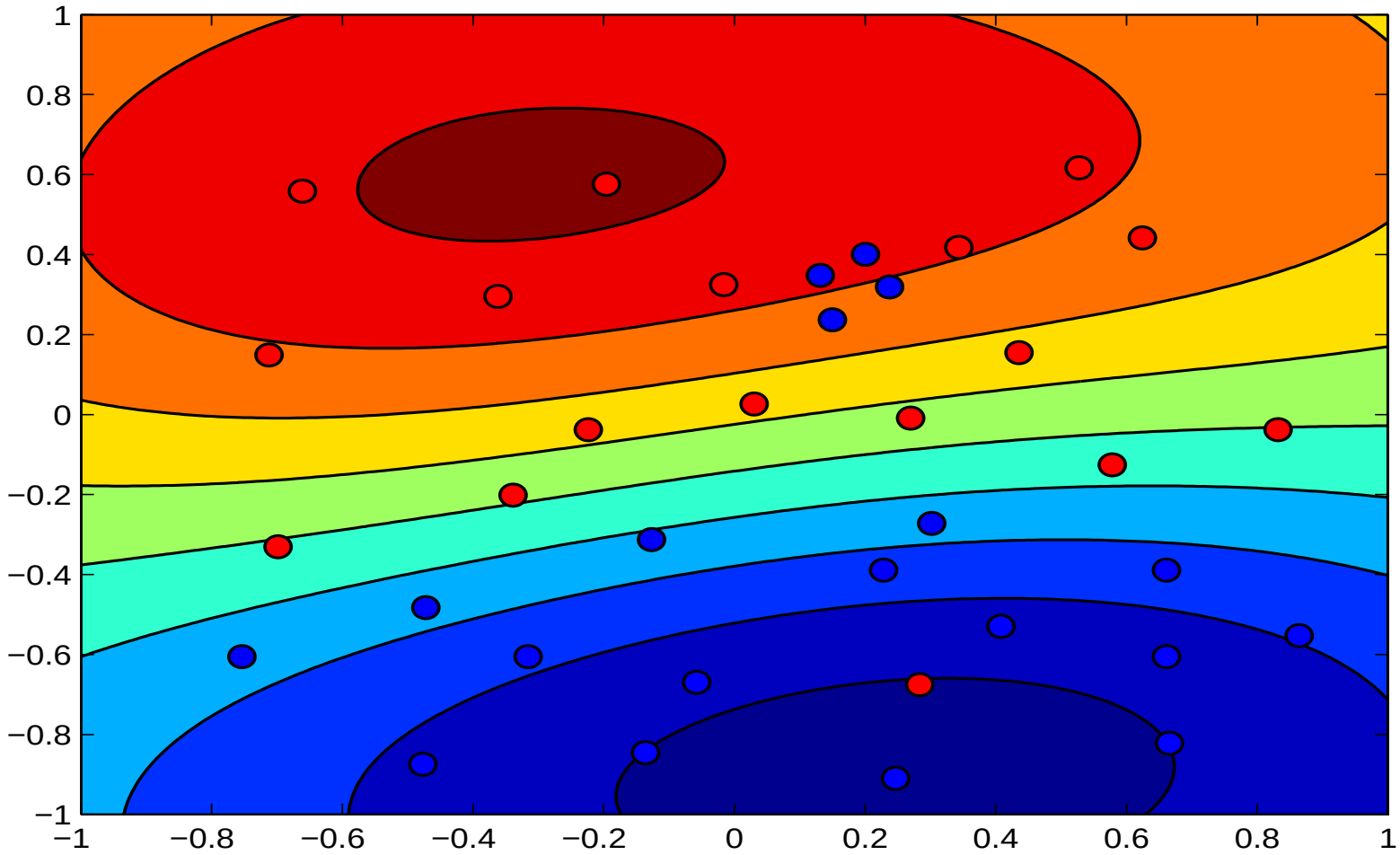


Example

What do we expect if we increase the value of λ?

Example

Consider a Gaussian kernel with $\sigma = 1$.
Regularization parameter $\lambda = 10$.

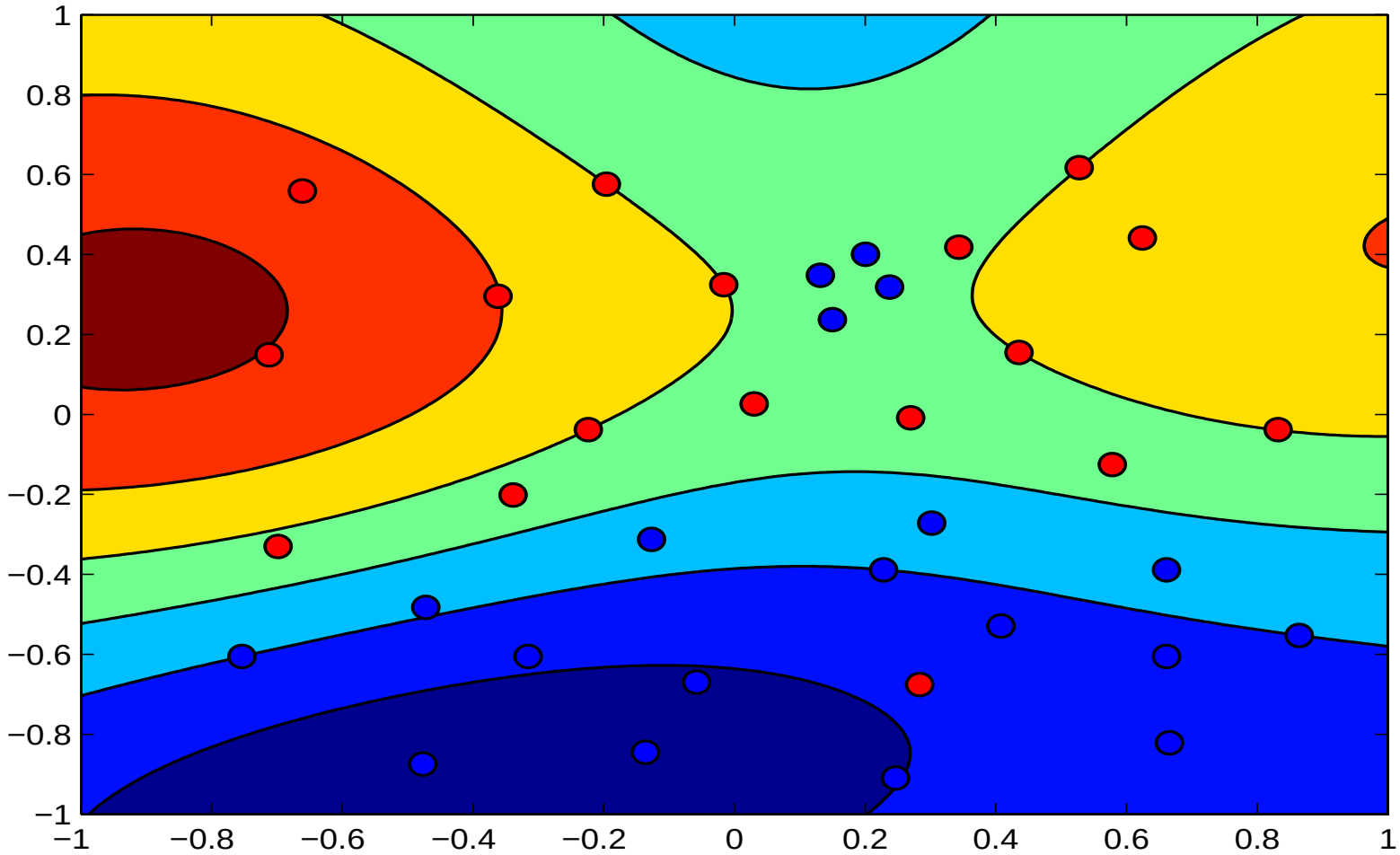


Example

What do we expect if we decrease the value of λ?

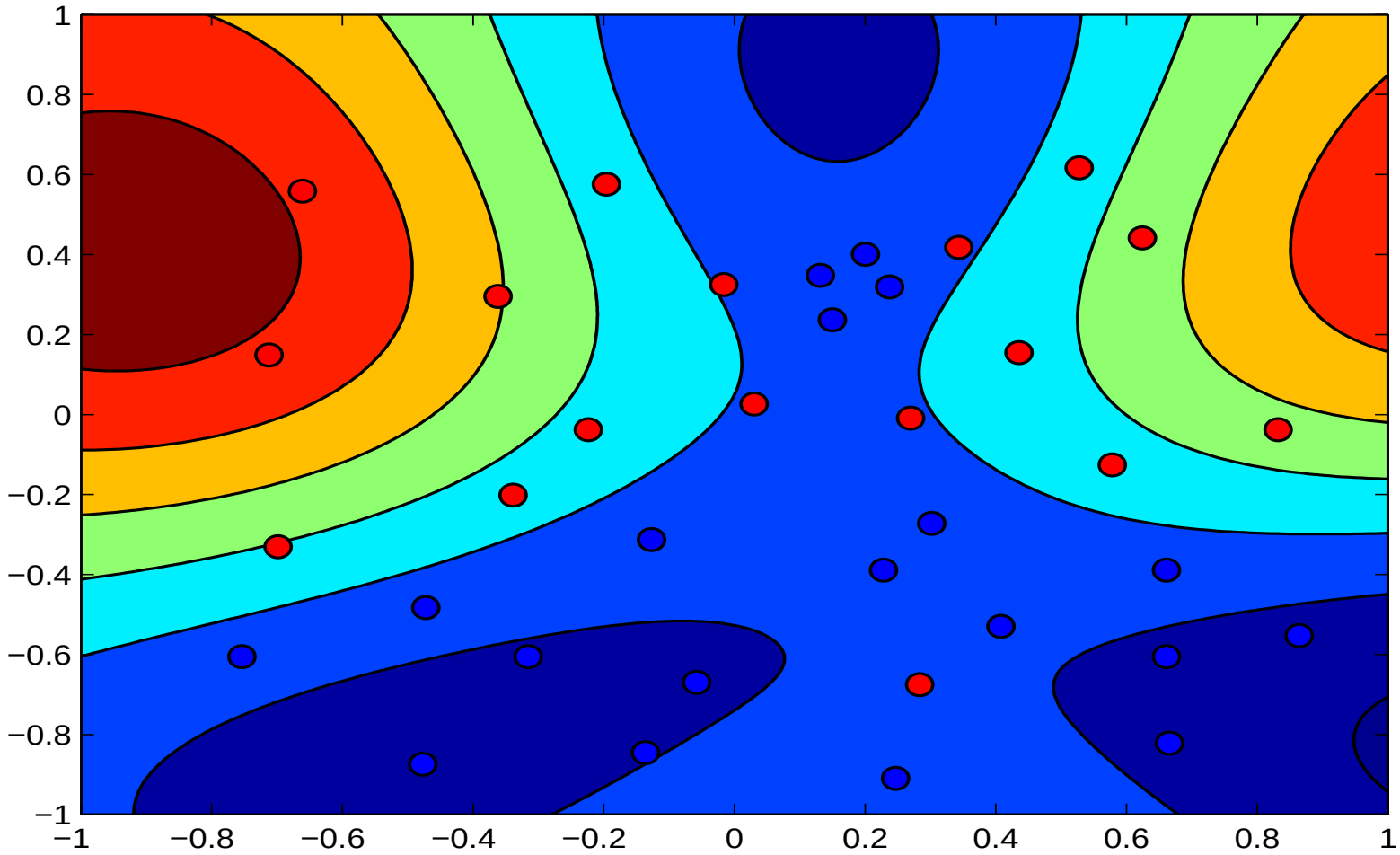
Example

Consider a Gaussian kernel with $\sigma = 1$.
Regularization parameter $\lambda = 0.1$.



Example

Consider a Gaussian kernel with $\sigma = 1$.
Regularization parameter $\lambda = 0.01$.



The kernel trick

Consider the Regularized Least Squares problem using a linear model. But, instead of the original input space consider a mapping $\phi(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ of the data into another feature space.

The problem now is

$$\alpha^* = \arg\min_{\alpha} \frac{1}{2} \sum_i (y_i - \alpha_i \phi(x_i))^2 + \frac{\lambda}{2} \|\alpha\|^2 =$$

$$= \arg\min_{\alpha} \frac{1}{2} (\mathbf{y} - \Phi\alpha)^T (\mathbf{y} - \Phi\alpha) + \frac{\lambda}{2} \alpha^T \alpha \implies$$

$$\implies \Phi^T \underbrace{(\mathbf{y} - \Phi\alpha)}_{\sigma} + \lambda\alpha = 0 \implies$$

$$\implies \alpha = \frac{1}{\lambda} \Phi^T \sigma = \frac{1}{\lambda} \sum_{t=1}^{N} \sigma_t \phi(\mathbf{x}_t)$$

Observe that the optimal alpha lies in the span of the feature vectors corresponding to the training examples.

The kernel trick

And the optimal σ ?

$$\sigma = (\mathbf{y} - \Phi\alpha) = (\mathbf{y} - \frac{1}{\lambda}\Phi\Phi^T\sigma)$$

$$\sigma = (I + \frac{1}{\lambda}\Phi\Phi^T)^{-1}\mathbf{y} = (\Phi\Phi^T + \lambda I)^{-1}\lambda\mathbf{y}$$

and the prediction model becomes

$$\hat{y} = \Phi\alpha = \frac{1}{\lambda}\Phi\Phi^T\sigma$$

Observe that this is the same solution as in the kernel method where the Gram matrix is the inner product matrix of the data set casted in the mapped space $K = \Phi\Phi^T$. Using kernels can be seen as mapping data in an arbitrarily high dimensional space defined by the Gram matrix.

The kernel trick

Consider the case of the polynomial quadratic kernel and assume $\mathbf{x} \in \mathbb{R}^2$:

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

$$= (1 + x_1 x_1' + x_2 x_2')^2 =$$

$$1 + 2x_1 x_1' + 2x_2 x_2' + (x_1 x_1')^2 + (x_2 x_2')^2 + 2x_1 x_1' x_2 x_2' \implies$$

$$\Phi(\mathbf{x}) = [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2]^T \in \mathbb{R}^6$$

- Using kernels can be seen as mapping data in an arbitrarily high dimensional space spanned by $\Phi(x)$.
- The use of kernels allows a compact implicit description of the space generated by $\Phi(x)$ by means of the Gram matrix $K(x, \cdot) = \Phi(x)\Phi(\cdot)$.
- Observe that $K \in \mathbb{R}^{N \times N}$.
- The dimensionality of $\Phi(x)$ for a gaussian kernel is ∞.

The kernel trick

Consider the case of the polynomial quadratic kernel and assume $\mathbf{x} \in \mathbb{R}^2$:

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

$$= (1 + x_1 x_1' + x_2 x_2')^2 =$$

$$1 + 2x_1 x_1' + 2x_2 x_2' + (x_1 x_1')^2 + (x_2 x_2')^2 + 2x_1 x_1' x_2 x_2' \implies$$

$$\Phi(\mathbf{x}) = [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2]^T \in \mathbb{R}^6$$

- Using kernels can be seen as mapping data in an arbitrarily high dimensional space spanned by $\Phi(x)$.
- The use of kernels allows a compact implicit description of the space generated by $\Phi(x)$ by means of the Gram matrix $K(x, \cdot) = \Phi(x)\Phi(\cdot)$.
- Observe that $K \in \mathbb{R}^{N \times N}$.
- The dimensionality of $\Phi(x)$ for a gaussian kernel is ∞.

A brief digression on the curse of dimensionality

Kernels will project data into presumably very large dimensional spaces. If overfitting is the most serious problems in machine learning, the next most serious problem is high dimensional space data (specially in Instance Based Models). This problem has the proper name of

the curse of dimensionality.

Why is dimensionality a problem?

Our low-dimensional intuition does not work in high dimensions.

Examples

- Normal distribution.
- Uniform distribution on a hypercube.
- Points on a hyper grid.
- Approximation of spheres by a cube.
- Volume of a hypersphere/hyperorange.

Kernel intuition video

Check the video

The kernel trick

Remember the relationship between IBL and linear models (e.g. perceptron and linear SVM)? The final classifier has the form

$$f(x) = \underbrace{\sum_{i=1}^{N} \nu_i y_i x_i^T x}_{\text{Instance Based Learning}}$$

The kernel trick

Remember the relationship between IBL and linear models (e.g. perceptron and linear SVM)? The final classifier has the form

$$f(x) = \underbrace{\sum_{i=1}^{N} \nu_i y_i x_i^T x}_{\text{Instance Based Learning}}$$

The kernel trick

The kernel trick replaces an inner product $\langle x, z \rangle$ for the corresponding inner product in the reproducing kernel Hilbert space $K(x, z)$.

Kernelized version of the classifier: $f(x) = \sum_{i=1}^{N} \nu_i y_i K(x_i, x)$

Dual perceptron

Algorithm:

If $y_i w^T \mathbf{x}_i < 0$: $\quad w^{l+1} = w^l + y_i \mathbf{x}_i$

Model: $f(x) = w^T x$

Observe that $w = \sum_{i=1}^{l} \alpha_i \mathbf{x}_i$ [1]

With respect to α the perceptron update rule becomes:

$$\sum_{i=1}^{l+1} \alpha_i \mathbf{x}_i = \sum_{i=1}^{l} \alpha_i \mathbf{x}_i + y_i \mathbf{x}_i \implies \alpha_i \leftarrow \alpha_i + y_i$$

[1] α_i represents the number of times $\mathbf{x}_i$ is selected in the model. It corresponds to the number of y_i to be taken into account. In literature, kernel preceptron update rule is usually written as $\alpha_i \leftarrow \alpha_i + 1$. This is perfectly correct if we consider the model to be $f(x) = \sum_{k=1}^{l} \alpha_k \langle \mathbf{x}_k \mathbf{x} \rangle$ and $w = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i$. However, this means to redefine notation in terms of kernels.

Dual perceptron

Algorithm:

If $y_i w^T \mathbf{x}_i < 0$: $\quad w^{l+1} = w^l + y_i \mathbf{x}_i$
Model: $f(x) = w^T x$

Observe that $w = \sum_{i=1}^{l} \alpha_i \mathbf{x}_i$ With respect to α the perceptron update rule becomes:

$$\sum_{i=1}^{l+1} \alpha_i \mathbf{x}_i = \sum_{i=1}^{l} \alpha_i \mathbf{x}_i + y_i \mathbf{x}_i \implies \alpha_i \leftarrow \alpha_i + y_i$$

Dual Algorithm:

If $y_i \sum_{k=1}^{l} \alpha_k \langle \mathbf{x}_k, \mathbf{x}_i \rangle < 0$: $\quad \alpha_i \leftarrow \alpha_i + y_i$
Model: $f(x) = \sum_{k=1}^{l} \alpha_k \langle \mathbf{x}_k, \mathbf{x} \rangle$

Kernel perceptron

Generalized algorithm:

If $y_i f(\mathbf{x}_i) < 0$: $\quad \alpha_i \leftarrow \alpha_i + y_i$

Model: $f(x)$

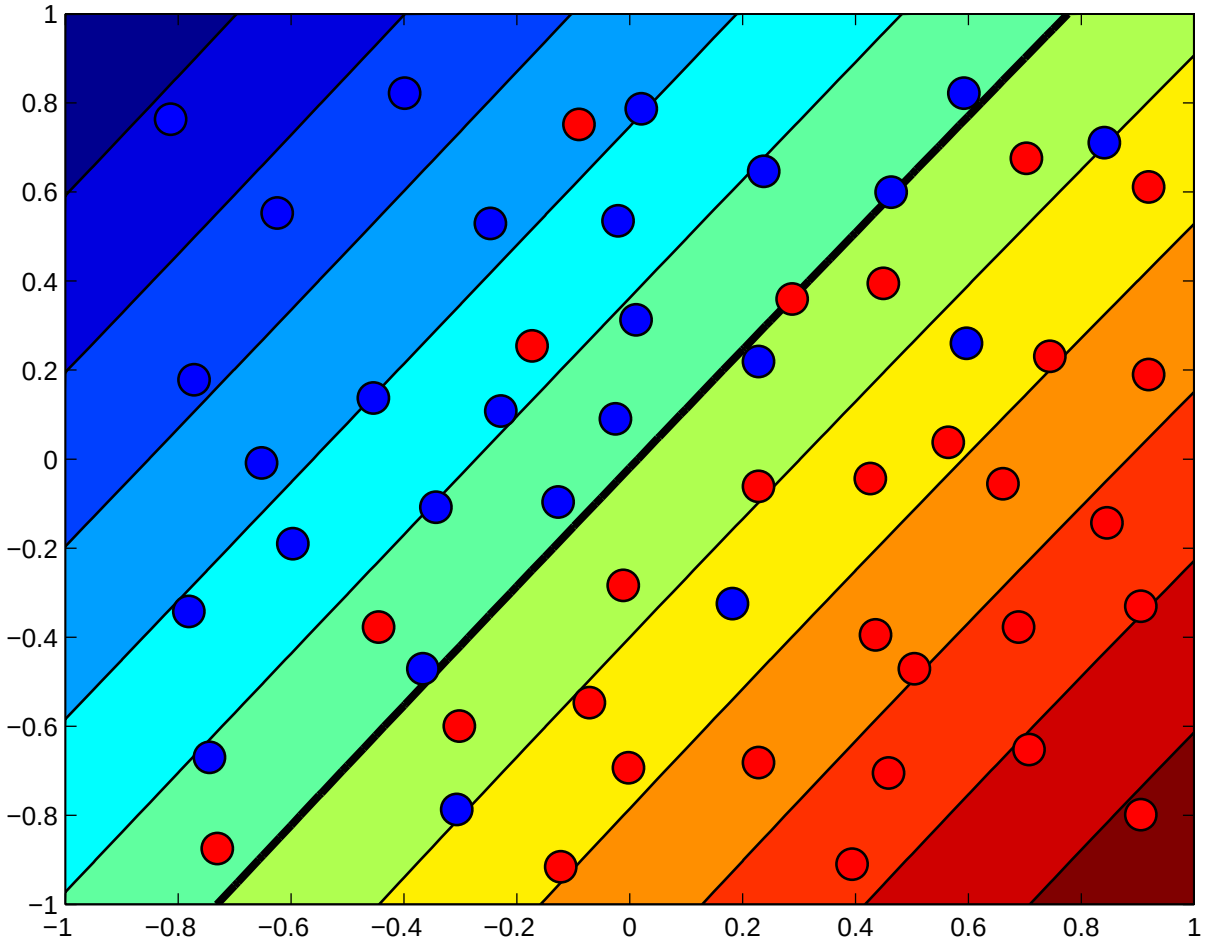
Remember that $f(x) = K(\mathbf{X}, x)^T \alpha$ in RKHS. Thus,

Perceptron kernel algorithm:

If $y_i \sum_{j=1}^{N} K(x_j, x_i)\alpha_j < 0$: $\quad \alpha_i \leftarrow \alpha_i + y_i$

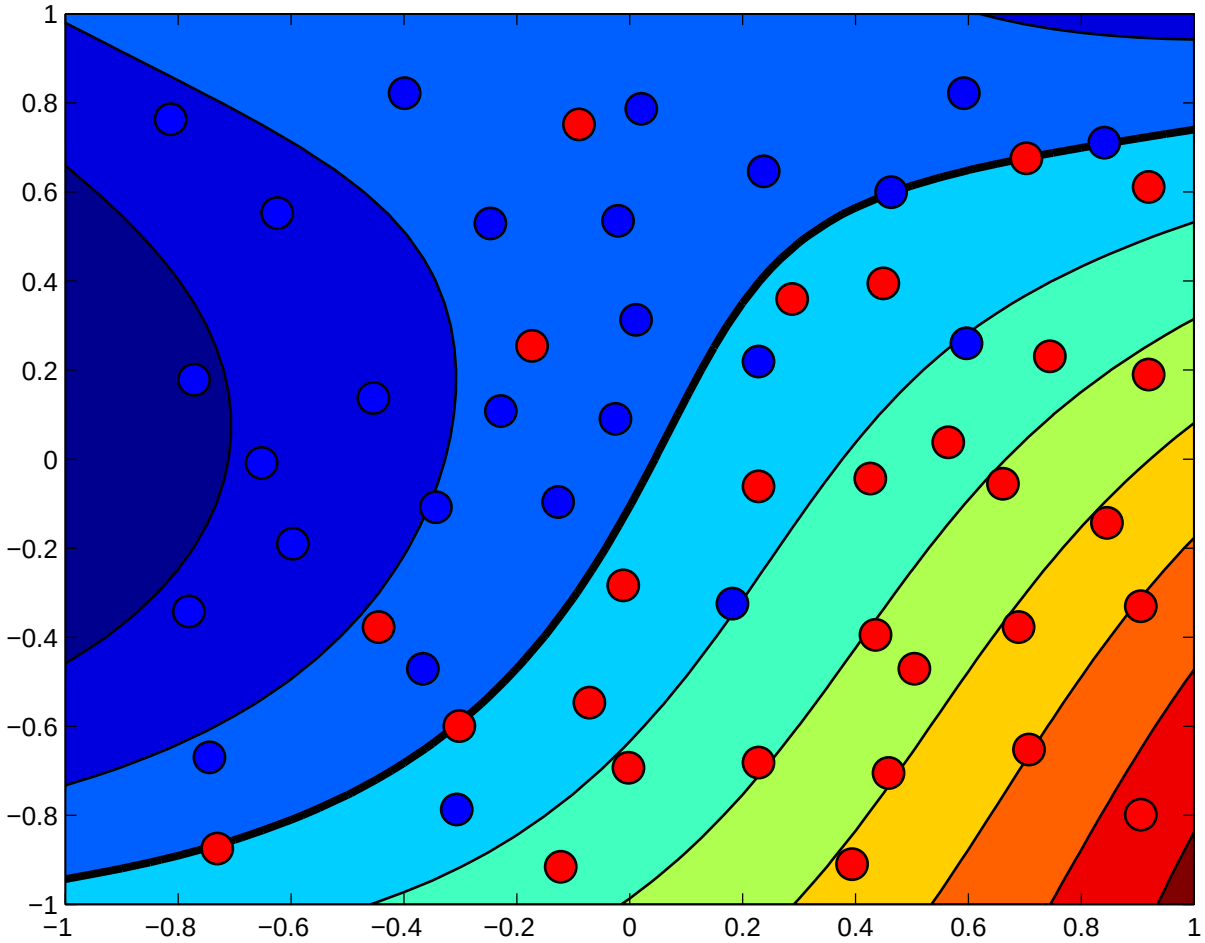
Model: $f(x) = \sum_{j=1}^{N} K(x_j, x)\alpha_j = K(\mathbf{X}, x)^T \alpha$

Kernel perceptron



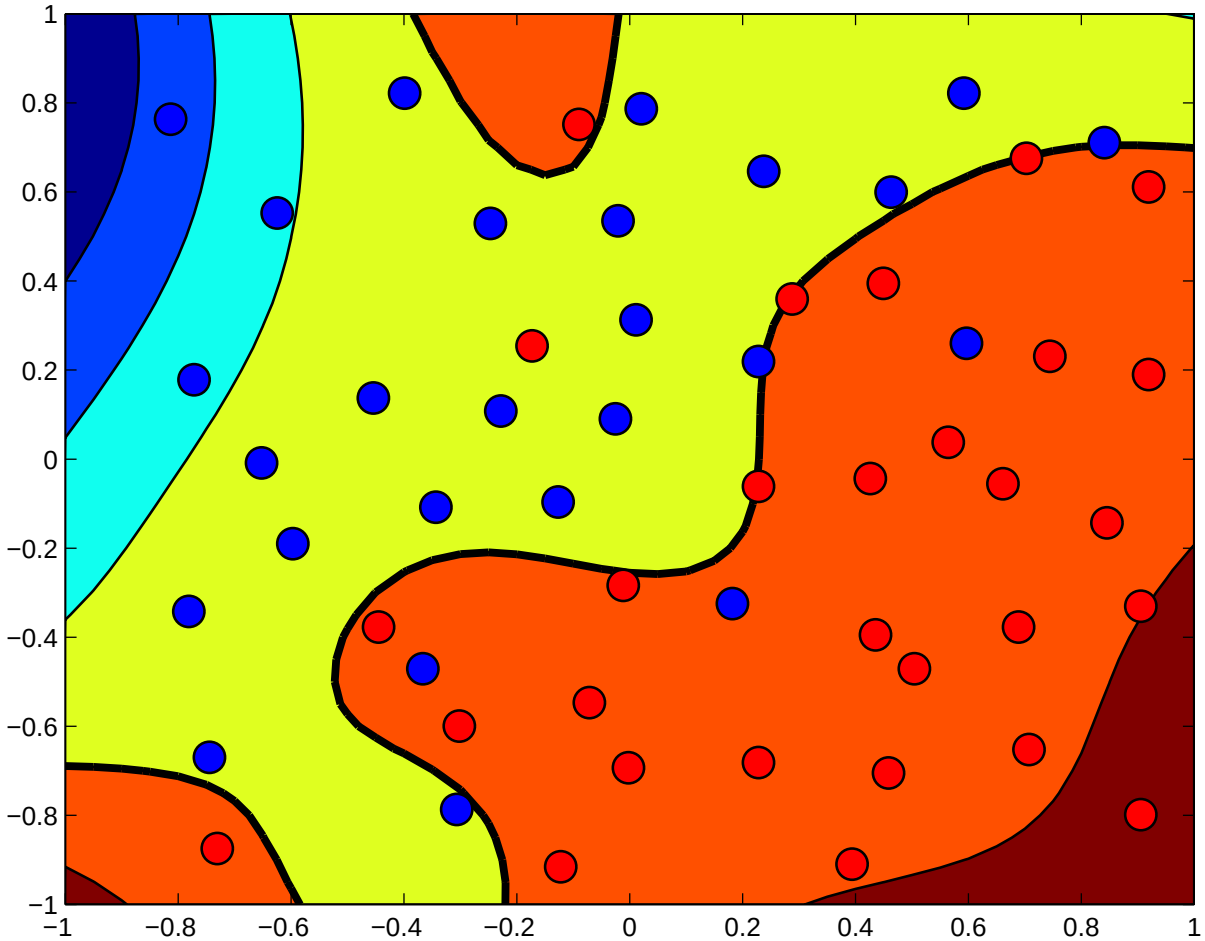
$(\sigma = 100)$

Kernel perceptron



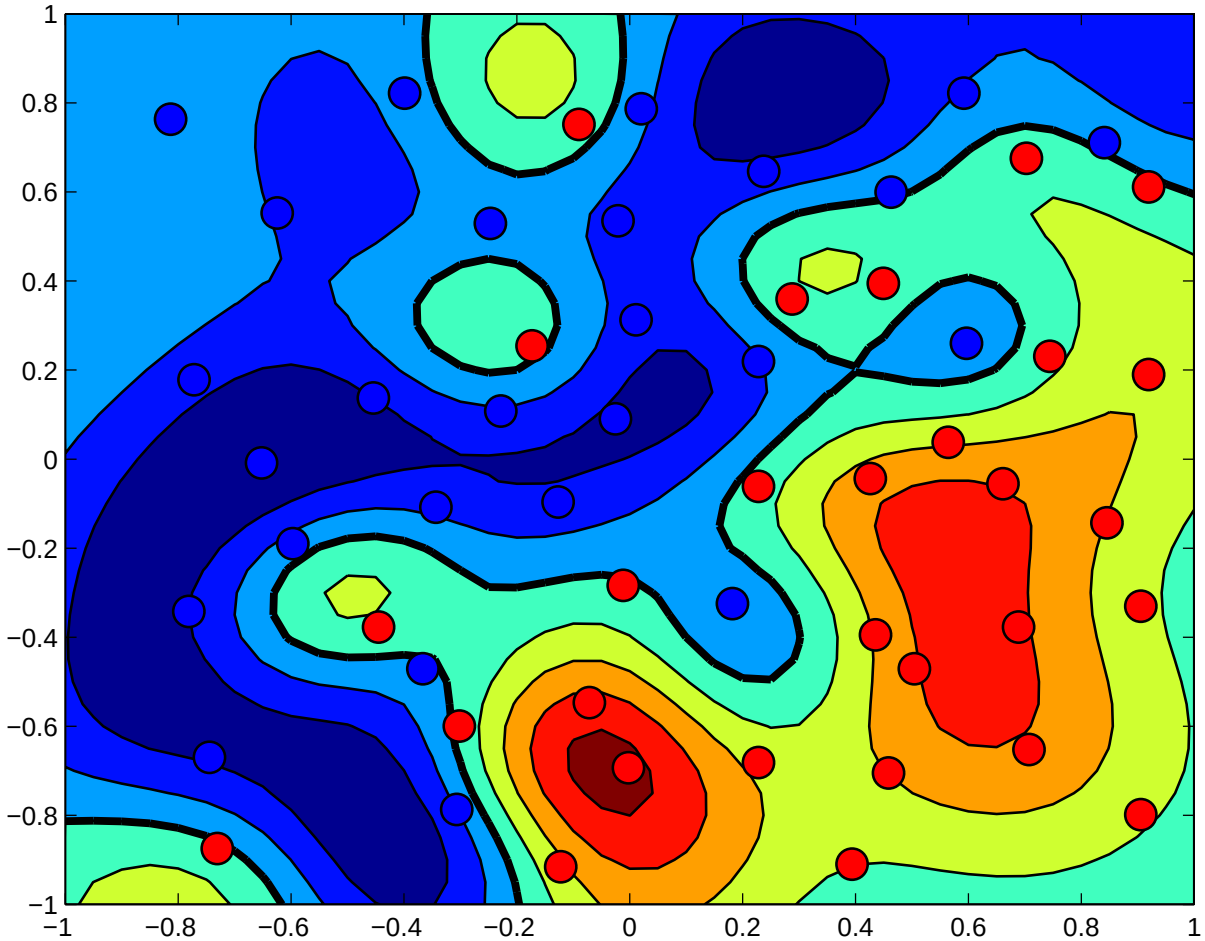
$(\sigma = 10)$

Kernel perceptron



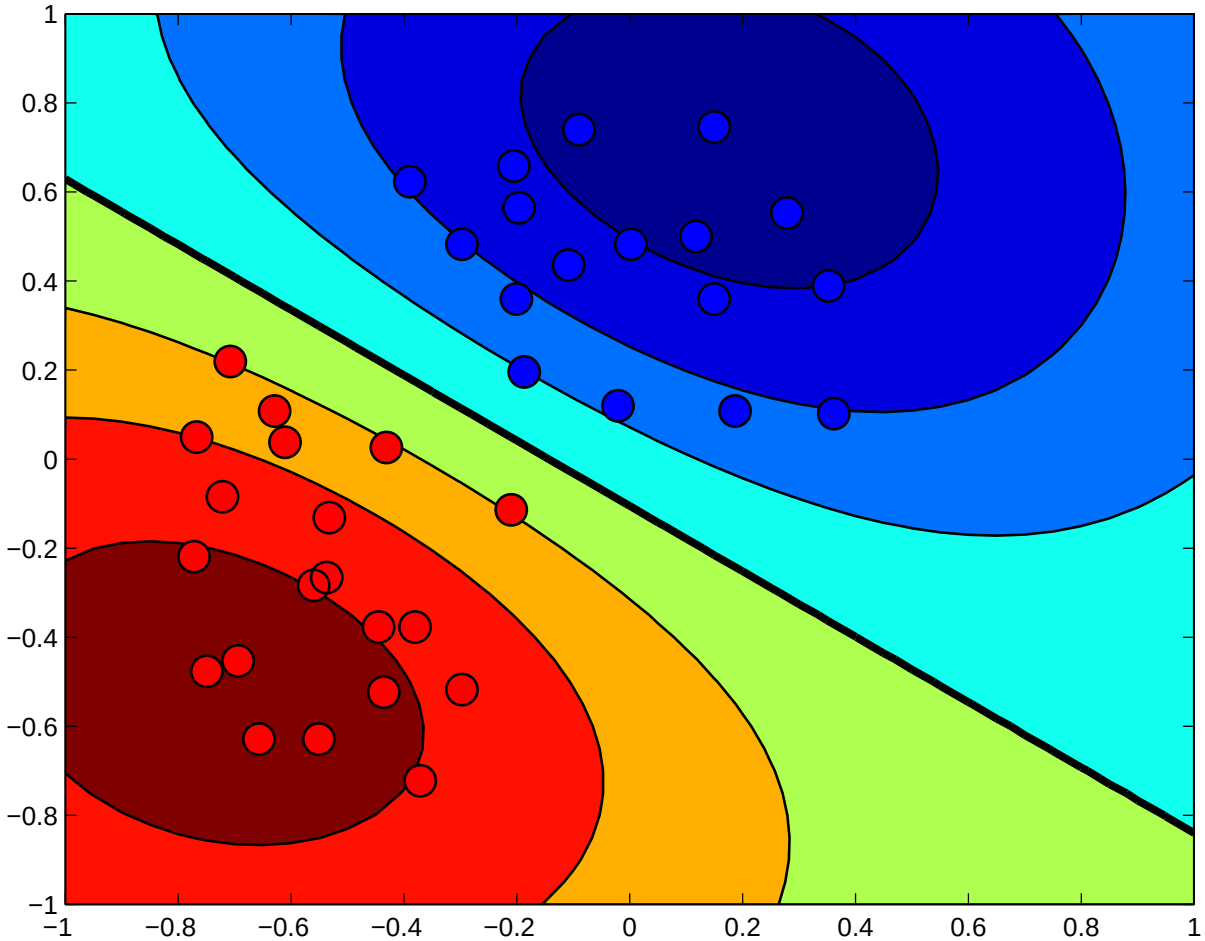
$(\sigma = 1)$

Kernel perceptron



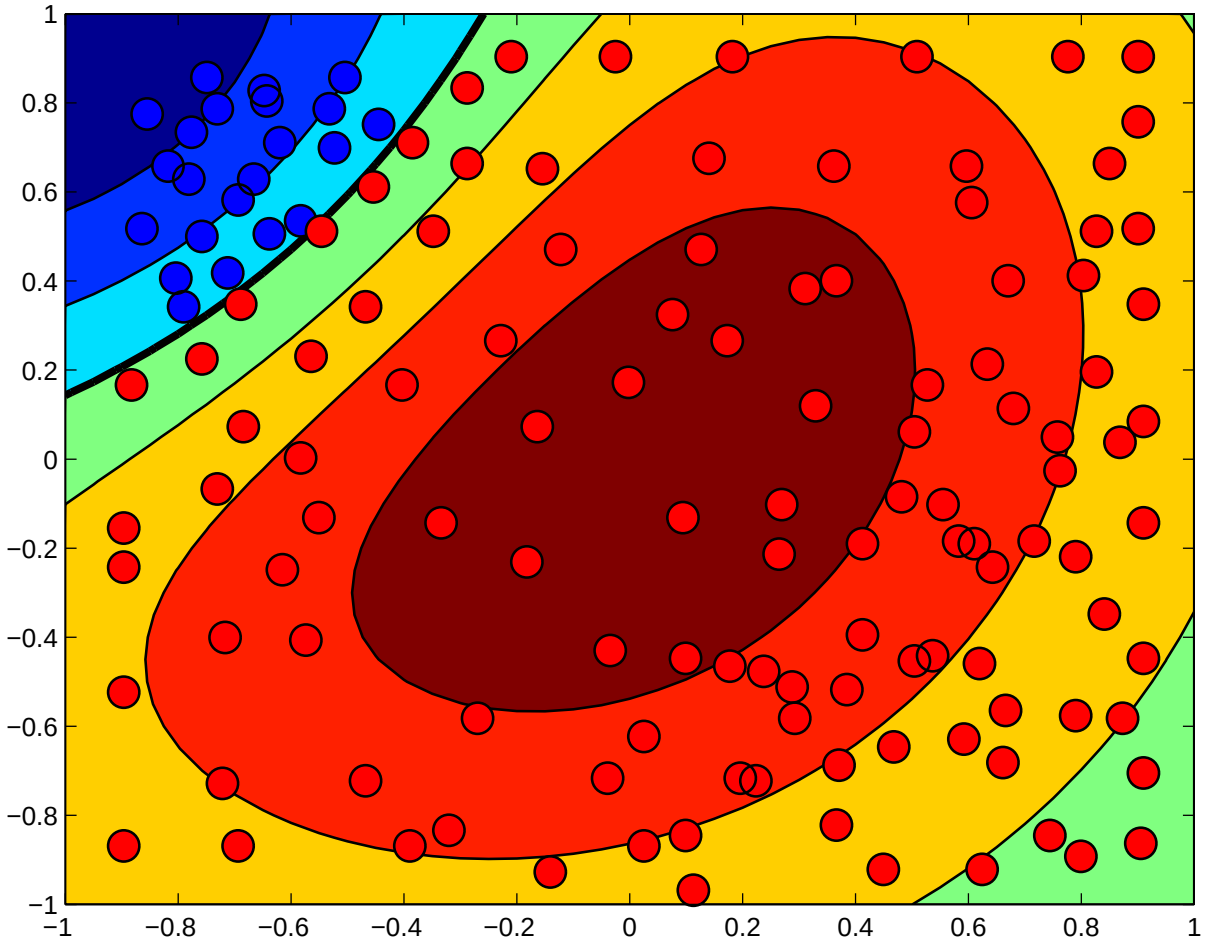
$(\sigma = 0.1)$

Kernel perceptron



$(\sigma = 1)$

Kernel perceptron



$(\sigma = 1)$

Online kernel perceptron

Observation Individual updates considering one data at a time are attractive since they naturally handle large scale data sets.

A naive online version of kernel perceptron:

Data: A pair $(\mathbf{x}, y)$

Result: A model $\{\alpha, sv\}$, sv the set of "support vectors", α the weighting vector

Compute the vector $K(x, sv)$;

if $yK\alpha < 0$ **then**

  if $x \in sv$ **then**

    $i \leftarrow \mathbf{idx}(x \in sv)$;

    $\alpha_i \leftarrow \alpha_i + y$;

  else

    $sv \leftarrow sv \cup x$;

    $\alpha \leftarrow \alpha \cup y$;

  end

end

Support vector machines ... AGAIN!

Recall the dual formulation of SVM:

$$\begin{array}{lll} \text{maximize} & \nu^T \mathbf{1} - \frac{1}{2}\nu^T Q \nu & \\ \text{subject to} & 0 \leqslant \nu_i \leqslant \lambda & \forall i = 1, \ldots, N \\ & \nu^T y = 0 & \end{array}$$

where $Q = \nu^T \mathbf{diag}(\mathbf{y}) X^T X \mathbf{diag}(\mathbf{y}) \nu$. We can kernelize it simply by setting:

$$Q = \nu^T \mathbf{diag}(\mathbf{y}) \mathbf{K} \mathbf{diag}(\mathbf{y}) \nu.$$

where $\mathbf{K}(i,j) = K(x_i, x_j)$. The associated solution becomes

$$f(x) = \sum_{j=1}^{N} K(x_j, x) y_j \nu_j$$

This is the classic formulation for Kernel SVM

Kernels in the primal!

The not so popular kernelized version of SVM is as follows
Remember the generalization of the norms and the Representer's theorem in RKHS, then we can rewrite SVM as follows,

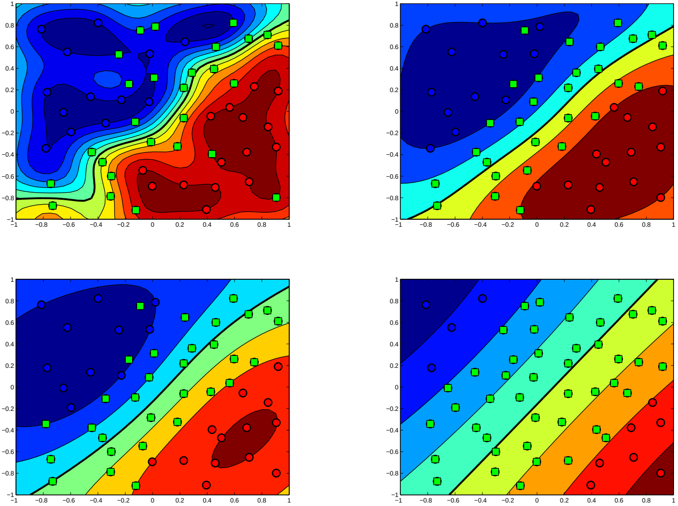
$$\begin{array}{ll} \text{minimize} & \|f\|_{\mathcal{H}} + \lambda u^T \mathbf{1} \\ \text{subject to} & y_i f(x_i) \geqslant 1 - u_i \quad \forall i = 1, \ldots, N \\ & u \geq 0 \end{array}$$

Using the former derivation, we can cast the problem as

$$\begin{array}{ll} \text{minimize} & a^T \mathbf{K} a + \lambda u^T \mathbf{1} \\ \text{subject to} & y_i a^T K(\mathbf{X}, x_i)^T \geqslant 1 - u_i \quad \forall i = 1, \ldots, N \\ & u \geq 0 \end{array}$$

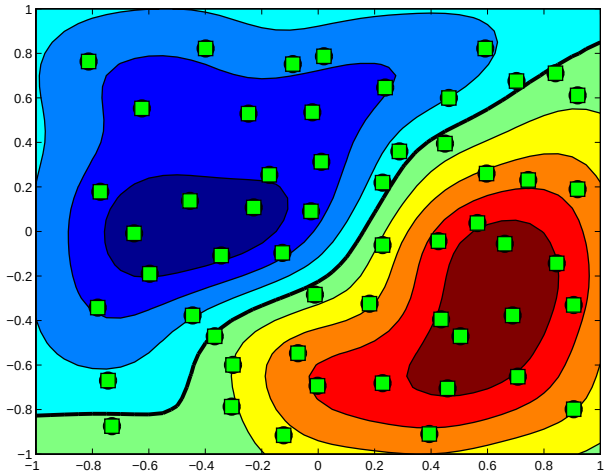
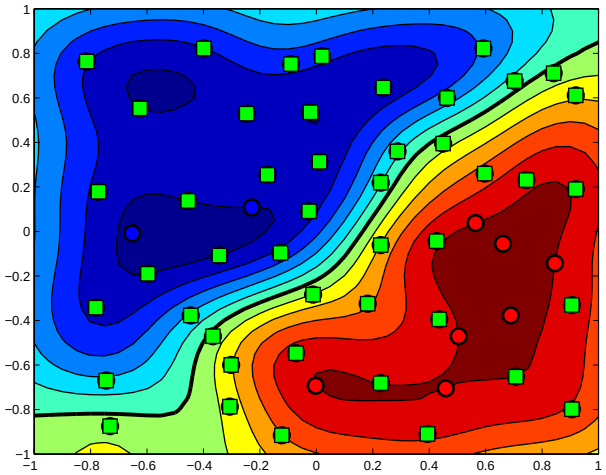
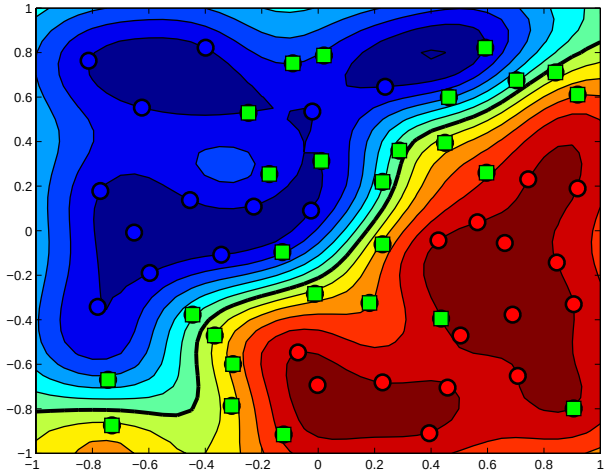
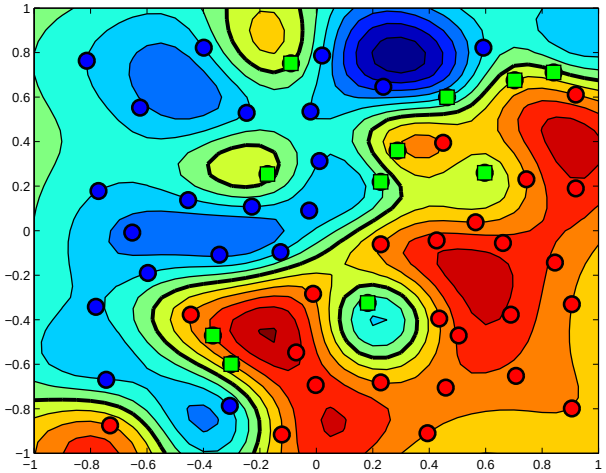
Support vector machines ... AGAIN!

Increasing $\sigma \in [0.1, 0.5, 1, 10]$ with $\lambda = 1$.



Support vector machines ... AGAIN!

Increasing $\lambda \in [0.1, 0.5, 1, 10]$ with $\sigma = 0.1$ (high expressivity).

Take home ideas

1. The notion of kernel allows to transform any linear classifier into a non-linear one.
2. Kernels are measures of similarity in a certain transformed space (usually high dimensional).
3. The kernel trick allows to kernelize any algorithm by replacing $\langle x, y \rangle$ by $K(x, y)$.
4. We can go beyond the kernel trick considering RKHS equivalents.