CS 446/ECE 449: Machine Learning

Lecture 6: Kernel Methods

Han Zhao
02/01/2024

ILLINOIS
Computer Science
GRAINGER COLLEGE OF ENGINEERING

Recap: Support Vector Machine

Max-margin principle: (Vapink' 82): choose w that maximizes the margin (distance to the closest data point)

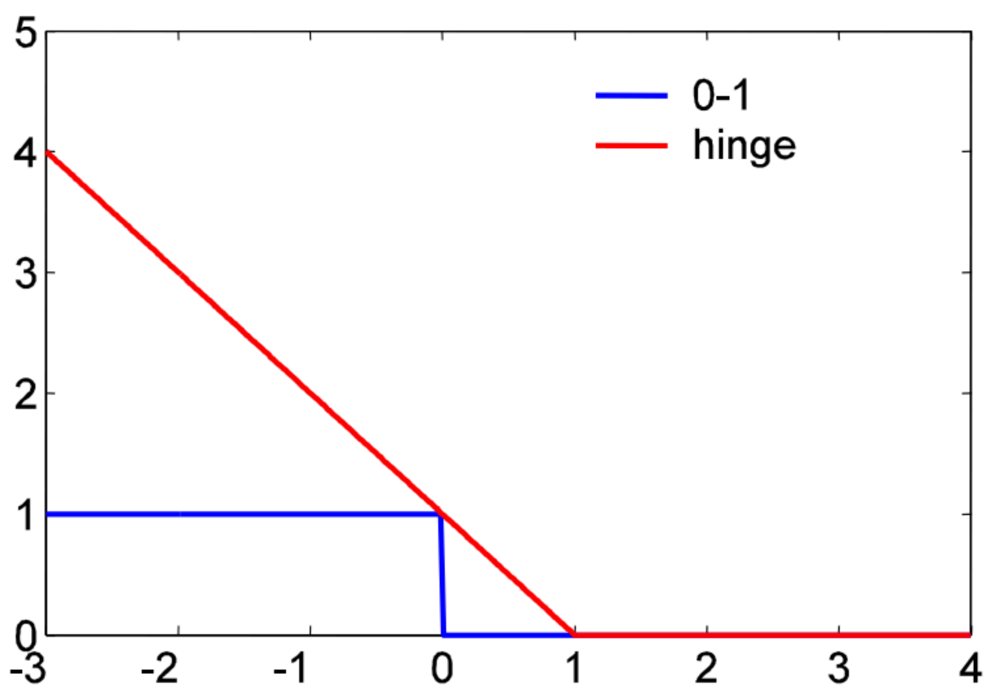
Support vector machines:

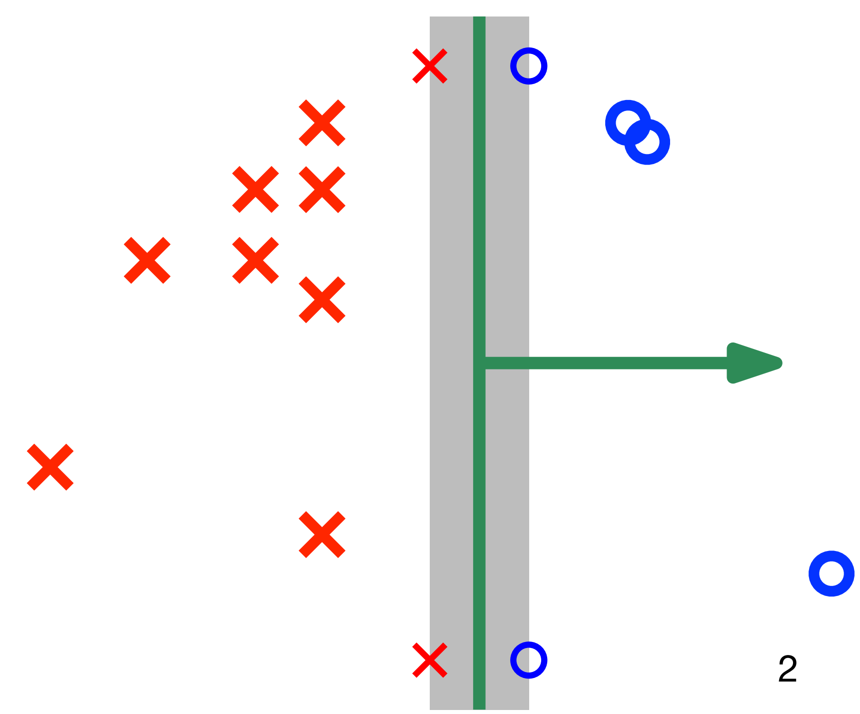
$$\min_{w \in \mathbb{R}^d} \sum_{i \in [n]} \ell_{\text{hinge}}(y^{(i)} \cdot w^\top x^{(i)}) + \frac{\lambda}{2} \|w\|_2^2,$$

where $\ell_{\text{hinge}}(t) := \max\{0, 1 - t\}$ is called the hinge-loss

Hinge loss

l_2 regularization of w





Recap: Support Vector Machine

Comparisons:

- E: supervised
- T: linear prediction
- P: zero-one, hinge, logistic, squared

Regularized linear regression (Ridge regression):

$$\min_{w \in \mathbb{R}^d} \sum_{i \in [n]} (y^{(i)} - w^\top x^{(i)})^2 + \frac{\lambda}{2} \|w\|_2^2,$$

Regularized logistic regression:

$$\min_{w \in \mathbb{R}^d} \sum_{i \in [n]} \ell_{\log}(y^{(i)} \cdot w^\top x^{(i)}) + \frac{\lambda}{2} \|w\|_2^2,$$

Support vector machines:

$$\min_{w \in \mathbb{R}^d} \sum_{i \in [n]} \ell_{\text{hinge}}(y^{(i)} \cdot w^\top x^{(i)}) + \frac{\lambda}{2} \|w\|_2^2,$$

Lecture Today

- Support Vector Machine (dual)
- Kernel Method

Support Vector Machine

Recall: given a linearly separable data for binary classification, our objective function of optimizing (hard-margin) SVM looks like follows:

$$\min_{w \in \mathbb{R}^d} \frac{1}{2}\|w\|_2^2, \quad \text{s.t.} \quad y^{(i)} w^\top x^{(i)} \geq 1, \forall i \in [n]$$

Note:

- This is an instance of the so-called "Quadratic Program", which belongs to convex problems
- Every convex program has a corresponding dual program
  - Clarifies the role of support vectors
  - Leads to a nice nonlinear approach: "kernel trick"
  - Gives another choice for optimization algorithms to solve for SVMs

Support Vector Machine

Recall: given a linearly separable data for binary classification, our objective function of optimizing (hard-margin) SVM looks like follows:

$$\min_{w \in \mathbb{R}^d} \quad \frac{1}{2}\|w\|_2^2$$
$$\text{s.t.} \quad y^{(i)} w^\top x^{(i)} \geq 1, \; \forall i \in [n]$$

How to obtain the corresponding dual program?

Key idea: introduce a dual variable $\alpha_i \geq 0$ for each of the constraint

- Interpretation of α_i: the "price" to pay if the corresponding constraint is violated
- With the dual variables, we can equivalently transform a constrained opt. to an unconstrained one

Support Vector Machine

Recall: given a linearly separable data for binary classification, our objective function of optimizing (hard-margin) SVM looks like follows:

$$\min_{w \in \mathbb{R}^d} \quad \frac{1}{2}\|w\|_2^2$$

$$\text{s.t.} \quad y^{(i)} w^\top x^{(i)} \geq 1, \ \forall i \in [n]$$

$$\min_{w \in \mathbb{R}^d} \max_{\alpha \in \mathbb{R}_+^n} \quad \frac{1}{2}\|w\|_2^2 + \sum_{i \in [n]} \alpha_i \left(1 - y^{(i)} w^\top x^{(i)}\right)$$

Claim: the optimal solutions of these two problems are the same (why?)

Support Vector Machine

Recall: given a linearly separable data for binary classification, our objective function of optimizing (hard-margin) SVM looks like follows:

$$\min_{w \in \mathbb{R}^d} \quad \frac{1}{2}\|w\|_2^2$$

$$\text{s.t.} \quad y^{(i)} w^\top x^{(i)} \geq 1, \ \forall i \in [n]$$

$$\min_{w \in \mathbb{R}^d} \max_{\alpha \in \mathbb{R}_+^n} \quad \frac{1}{2}\|w\|_2^2 + \sum_{i \in [n]} \alpha_i \left(1 - y^{(i)} w^\top x^{(i)}\right)$$

Let's consider two cases:

- If the i-th constraint holds, i.e., $1 - y^{(i)} w^\top x^{(i)} \leq 0$, then $\alpha_i^* = 0$
- If the i-th constraint is violated, i.e., $1 - y^{(i)} w^\top x^{(i)} > 0$, then $\alpha_i^* \to \infty$

Support Vector Machine

The Lagrangian $\mathscr{L}(w, \alpha)$:

$$\min_{w \in \mathbb{R}^d} \max_{\alpha \in \mathbb{R}^n_+} \quad \mathscr{L}(w, \alpha) := \frac{1}{2}\|w\|_2^2 + \sum_{i \in [n]} \alpha_i \left(1 - y^{(i)} w^\top x^{(i)}\right)$$

The dual variables α_i are also called the Lagrange multipliers

In general, for an arbitrary function $f(x, y)$, we have the following relationship holds, known as "weak duality":

$$\min_x \max_y f(x, y) \geq \max_y \min_x f(x, y)$$

Can understand this inequality from a game-theoretic perspective:

- There are two players x, y for a one-shot, zero-sum game, with payoff $f(x, y)$
- Player x would like to minimize the payoff
- Player y would like to maximize the payoff
- LHS = Player x goes first then player y
- The minimax inequality holds due to "second-mover advantage"

Support Vector Machine

The Lagrangian $\mathscr{L}(w, \alpha)$:

$$\min_{w \in \mathbb{R}^d} \max_{\alpha \in \mathbb{R}^n_+} \quad \mathscr{L}(w, \alpha) := \frac{1}{2}\|w\|_2^2 + \sum_{i \in [n]} \alpha_i \left(1 - y^{(i)} w^\top x^{(i)}\right)$$

The dual variables α_i are also called the Lagrange multipliers

In general, for an arbitrary function $f(x, y)$, we have the following relationship holds, known as "weak duality":

$$\min_x \max_y f(x, y) \geq \max_y \min_x f(x, y)$$

For convex problems with affine constraints, "strong duality" holds:

$$\min_x \max_y f(x, y) = \max_y \min_x f(x, y)$$

Hence,

$$\min_{w \in \mathbb{R}^d} \max_{\alpha \in \mathbb{R}^n_+} \mathscr{L}(w, \alpha) = \max_{\alpha \in \mathbb{R}^n_+} \min_{w \in \mathbb{R}^d} \mathscr{L}(w, \alpha)$$

Support Vector Machine

The Lagrangian $\mathscr{L}(w, \alpha)$:

$$\min_{w \in \mathbb{R}^d} \max_{\alpha \in \mathbb{R}^n_+} \quad \mathscr{L}(w, \alpha) := \frac{1}{2}\|w\|_2^2 + \sum_{i \in [n]} \alpha_i \left(1 - y^{(i)} w^\top x^{(i)}\right)$$

The dual variables α_i are also called the Lagrange multipliers
We can then define the following primal and dual problems:

- Primal problem: $P(w) := \max_{\alpha \in \mathbb{R}^n_+} \mathscr{L}(w, \alpha)$
- Dual problem: $D(\alpha) := \min_{w \in \mathbb{R}^d} \mathscr{L}(w, \alpha)$

By strong duality, we have

$$\min_{w \in \mathbb{R}^d} P(w) = \max_{\alpha \in \mathbb{R}^n_+} D(\alpha)$$

Support Vector Machine

The dual problem $D(\alpha)$:

$$\min_{w \in \mathbb{R}^d} \frac{1}{2}\|w\|_2^2 + \sum_{i \in [n]} \alpha_i \left(1 - y^{(i)} w^\top x^{(i)}\right)$$

For any fixed $\alpha \in \mathbb{R}_+^n$, we can first solve the internal optimization problem w.r.t. w, which is an unconstrained quadratic problem.

Setting the gradient to 0:

$$\nabla_w \left(\frac{1}{2}\|w\|_2^2 + \sum_{i \in [n]} \alpha_i \left(1 - y^{(i)} w^\top x^{(i)}\right) \right) = 0$$

we have

$$w = \sum_{i \in [n]} \alpha_i y^{(i)} x^{(i)}$$

Support Vector Machine

The dual problem $D(\alpha)$:

$$D(\alpha) = \min_{w \in \mathbb{R}^d} \frac{1}{2}\|w\|_2^2 + \sum_{i \in [n]} \alpha_i \left(1 - y^{(i)} w^\top x^{(i)}\right)$$

Plugging $w = \sum_{i \in [n]} \alpha_i y^{(i)} x^{(i)}$ into the above dual problem, we have

$$D(\alpha) = \sum_{i \in [n]} \alpha_i - \frac{1}{2} \sum_{i,j \in [n]} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)\top} x^{(j)}$$

$$= \mathbf{1}_n^\top \alpha - \frac{1}{2} \alpha^\top K \alpha$$

where $\mathbf{1}_n \in \mathbb{R}^n$ is a all-one vector of dim-n, and $K \in \mathbb{R}_+^{n \times n}$ with $K_{ij} := \left(y^{(i)} x^{(i)}\right)^\top \left(y^{(j)} x^{(j)}\right)$.

Support Vector Machine

The dual problem $D(\alpha)$:

$$\max_{\alpha \in \mathbb{R}_+^n} D(\alpha) = \sum_{i \in [n]} \alpha_i - \frac{1}{2} \sum_{i,j \in [n]} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)\top} x^{(j)} = \mathbf{1}_n^\top \alpha - \frac{1}{2} \alpha^\top K \alpha$$

Note:

- The dual optimization problem w.r.t. α is still a quadratic program
- In the primal problem $P(w)$, $w \in \mathbb{R}^d$ is the optimization variable
- In the dual problem $D(\alpha)$, $\alpha \in \mathbb{R}_+^n$ is the optimization variable
- Both the primal and the dual problems have affine constraints
- Similar to the primal problem, we can use off-the-shelf convex solvers to find the optimal α^*

Once we have the optimal α^*, we can recover the optimal w^* with

$$w^* = \sum_{i \in [n]} \alpha_i^* y^{(i)} x^{(i)}$$

Support Vector Machine

The dual problem $D(\alpha)$:

$$\max_{\alpha \in \mathbb{R}^n_+} D(\alpha) = \sum_{i \in [n]} \alpha_i - \frac{1}{2} \sum_{i,j \in [n]} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)\top} x^{(j)} = \mathbf{1}_n^\top \alpha - \frac{1}{2} \alpha^\top K \alpha$$

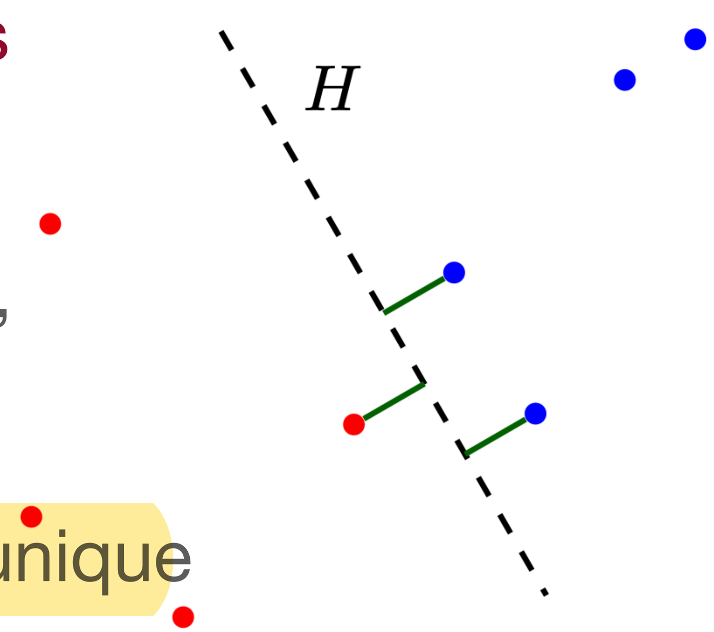
Once we have the optimal α^*, we can recover the optimal w^* with

$$w^* = \sum_{i \in [n]} \alpha_i^* y^{(i)} x^{(i)}$$

- The optimal normal vector w^* is a linear combination of $y^{(i)} x^{(i)}$
- Only the ones with $\alpha_i^* > 0$ contributes to w^*
- The point $y^{(i)} x^{(i)}$ with $\alpha_i^* > 0$ are called support vectors
- In fact, with $\alpha_i^* > 0$, we must have $y^{(i)} w^\top x^{(i)} = 1$

(due to the so-called complementary slackness condition), which coincides with our geometric definition of support vectors as well.

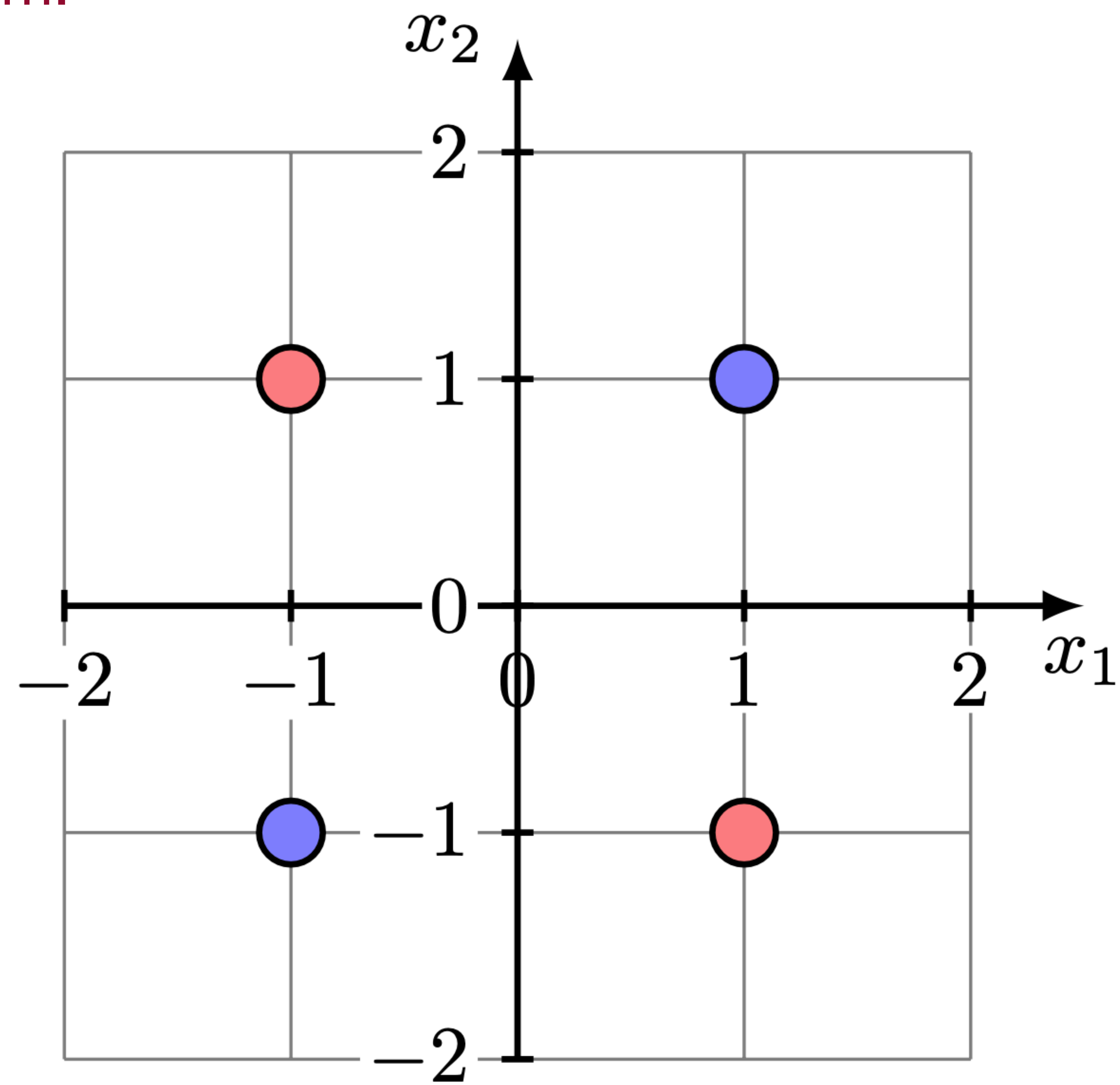
- Dual solutions and support vectors are not necessarily unique (even if the primal solution is unique)



Kernel Method

But, the dual problem formulation is still a hard-margin linear SVM

The XOR problem:



Think: not possible to perfectly classify the XOR problem with linear predictors

Kernel Method

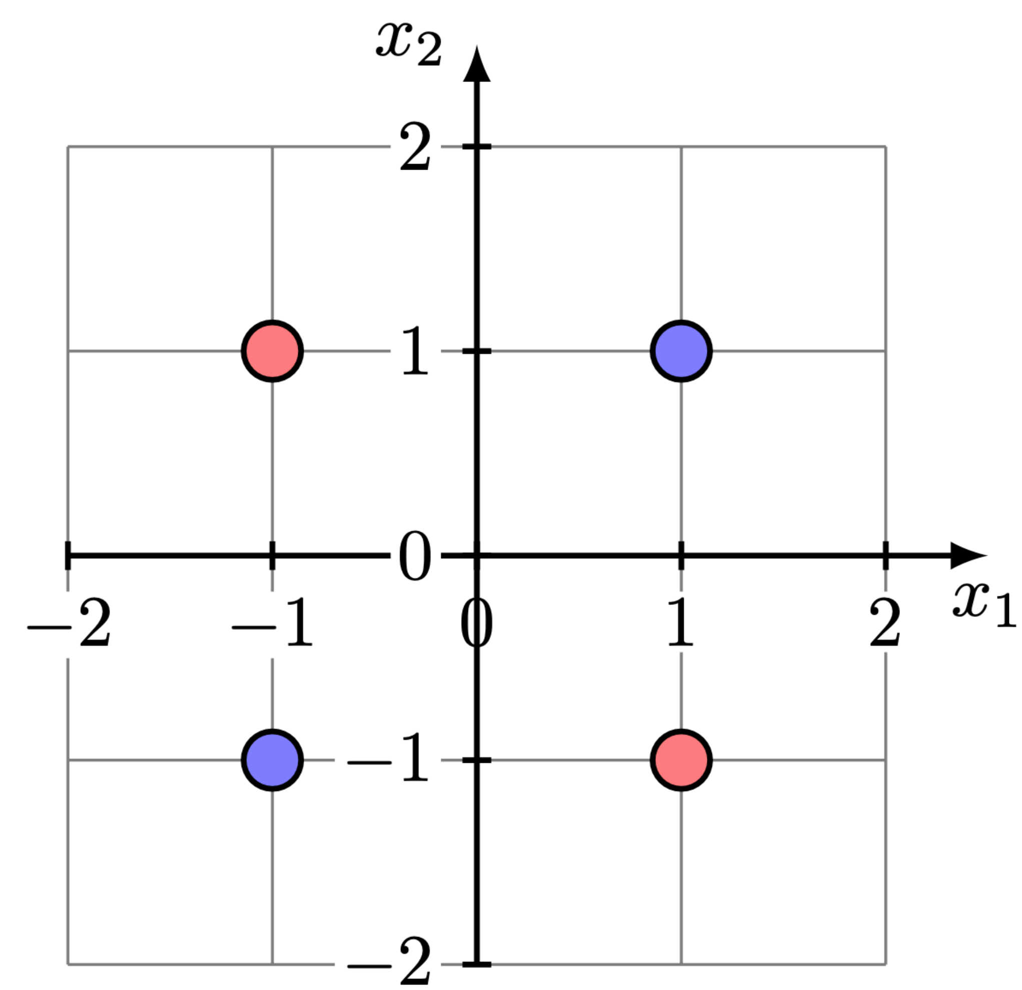
Key idea: feature mapping/lifting

$$\phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \to (x_1, x_2, x_1 x_2)$$

Under this feature map $\phi(\cdot)$, the XOR problem becomes:

Finding a linear classifier $w \in \mathbb{R}^3$ that correctly predicts the following 4 points:

- $(1,1,1)$
- $(1,-1,-1)$
- $(-1,1,-1)$
- $(-1,-1,1)$



One potential solution: $w^* = (0,0,1)$

Kernel Method

Key idea: feature map $\phi(\cdot) : \mathbb{R}^d \to \mathbb{R}^p$

The primal optimization problem of hard-margin SVM under ϕ:

$$\min_{w \in \mathbb{R}^d} \quad \frac{1}{2}\|w\|_2^2$$

$$\text{s.t.} \quad y^{(i)} w^\top \phi(x^{(i)}) \geq 1, \ \forall i \in [n]$$

Now the search space has p dimensions, and potentially $p \gg d$. In the case of $p = \infty$, we cannot solve the primal explicitly. How about the dual?

$$\max_{\alpha \in \mathbb{R}_+^n} D(\alpha) = \sum_{i \in [n]} \alpha_i - \frac{1}{2} \sum_{i,j \in [n]} \alpha_i \alpha_j y^{(i)} y^{(j)} \phi(x^{(i)})^\top \phi(x^{(j)})$$

$$= \mathbf{1}_n^\top \alpha - \frac{1}{2} \alpha^\top K \alpha$$

where $\mathbf{1}_n \in \mathbb{R}^n$ is a all-one vector of dim-n, and $K \in \mathbb{R}_+^{n \times n}$ with $K_{ij} := \left(y^{(i)} \phi(x^{(i)})\right)^\top \left(y^{(j)} \phi(x^{(j)})\right)$.

Kernel Method

Key idea: feature map $\phi(\,\cdot\,) : \mathbb{R}^d \to \mathbb{R}^p$

Dual form of hard-margin SVM under the feature map ϕ:

$$\max_{\alpha \in \mathbb{R}^n_+} D(\alpha) = \sum_{i \in [n]} \alpha_i - \frac{1}{2} \sum_{i,j \in [n]} \alpha_i \alpha_j y^{(i)} y^{(j)} \phi(x^{(i)})^\top \phi(x^{(j)})$$

$$= \mathbf{1}_n^\top \alpha - \frac{1}{2} \alpha^\top K \alpha$$

- The dual form never needs $\phi(x) \in \mathbb{R}^p$ explicitly, but only $\phi(x)^\top \phi(x') \in \mathbb{R}$
- Kernel trick: replace every $\phi(x)^\top \phi(x')$ with kernel evaluation $k(x, x')$
- Sometimes, $k(x, x')$ is much cheaper than $\phi(x)^\top \phi(x')$
- The idea started with SVM, but appears in many other linear models as well
- Downside: we need to explicitly maintain the kernel matrix $K \in \mathbb{R}^{n \times n}$, which could be expensive if n is large

Kernel Method

Key idea: feature map $\phi(\,\cdot\,) : \mathbb{R}^d \to \mathbb{R}^p$

Kernel example: affine features $\phi : \mathbb{R}^d \to \mathbb{R}^{d+1}$ with

$$\phi(x) = (1, x_1, \ldots, x_d)$$

Kernel form:

$$k(x, x') = \phi(x)^\top \phi(x') = 1 + x^\top x'$$

Kernel Method

Key idea: feature map $\phi(\,\cdot\,) : \mathbb{R}^d \to \mathbb{R}^p$

Kernel example: quadratic features $\phi : \mathbb{R}^d \to \mathbb{R}^p$ with

HW1: $\phi(x) = ?$

Kernel form:

$$k(x, x') = \phi(x)^\top \phi(x') = \left(1 + x^\top x'\right)^2$$

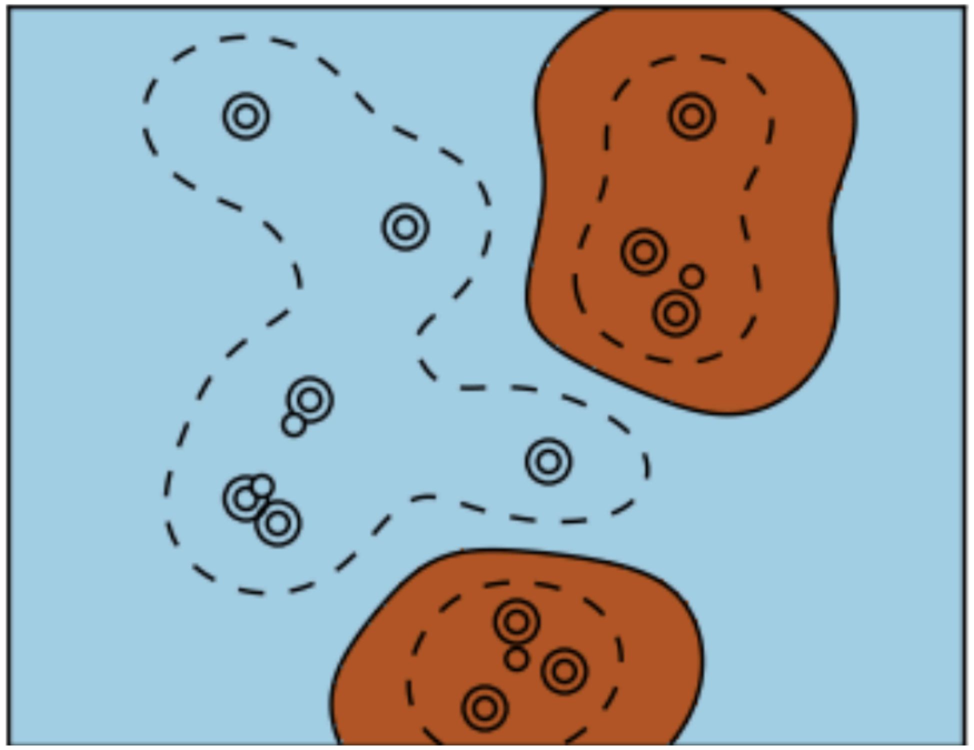
Kernel Method

Radial Basis Function kernel (RBF kernel, Gaussian kernel):

For any $\sigma > 0$, there is an infinite-dim feature map $\phi : \mathbb{R}^d \to \mathbb{R}^\infty$ such that

$$k(x, x') = \phi(x)^\top \phi(x') = \exp\left(-\frac{\|x - x'\|_2^2}{2\sigma^2}\right)$$

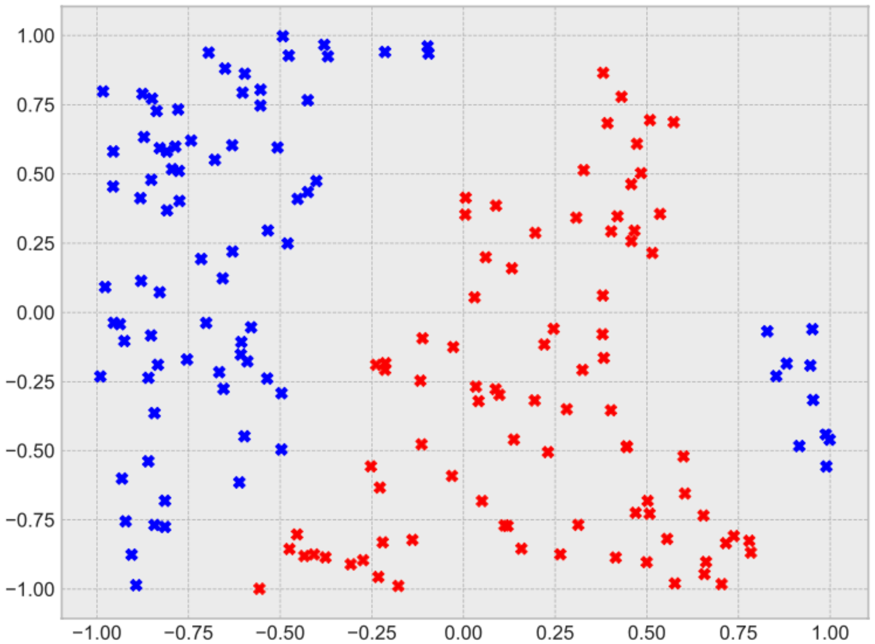
Note: despite the infinite-dim expansion, the kernel evaluation could be computed in $O(d)$ time



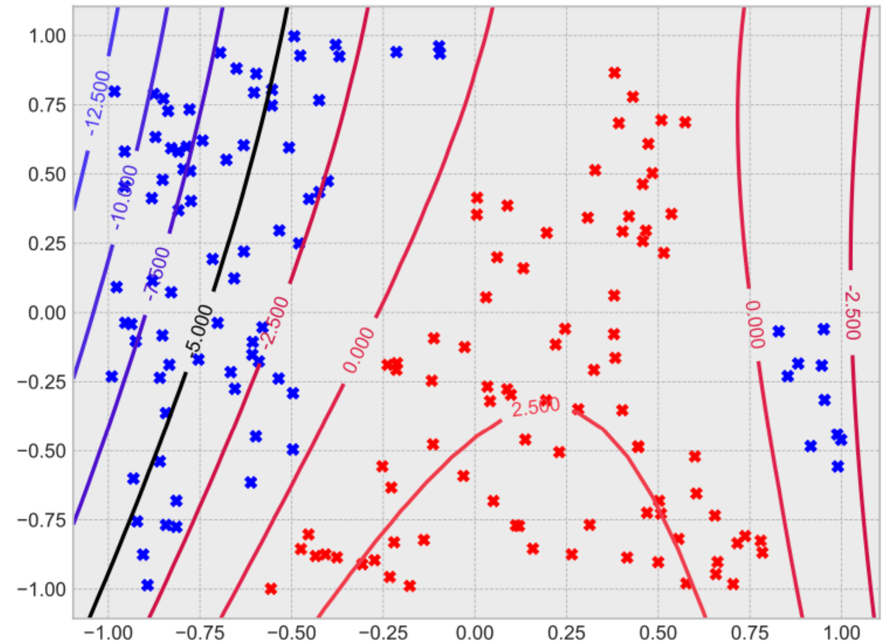
Intuition: kernel computes the similarity between data points

Kernel Method

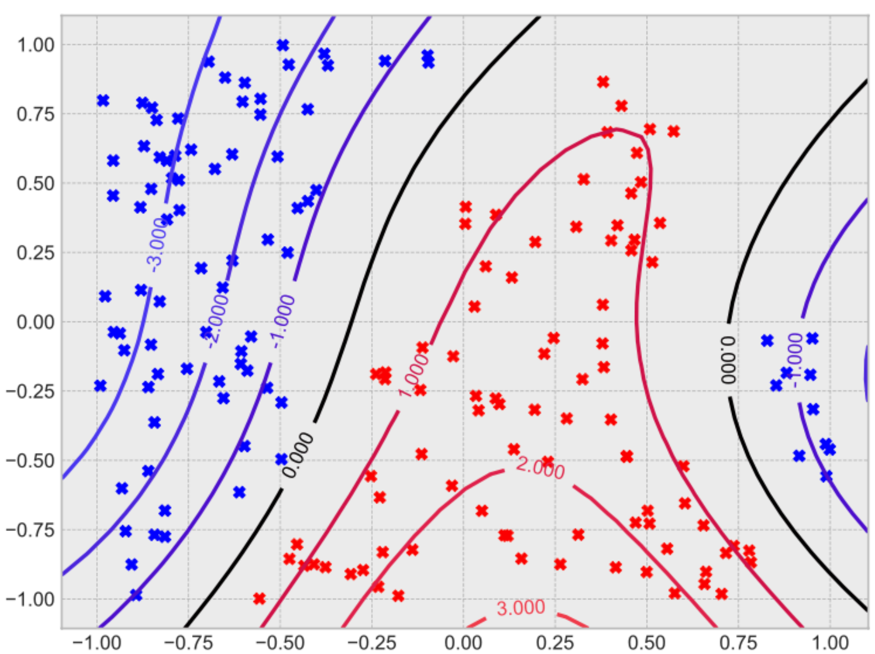
Radial Basis Function kernel (RBF kernel, Gaussian kernel):



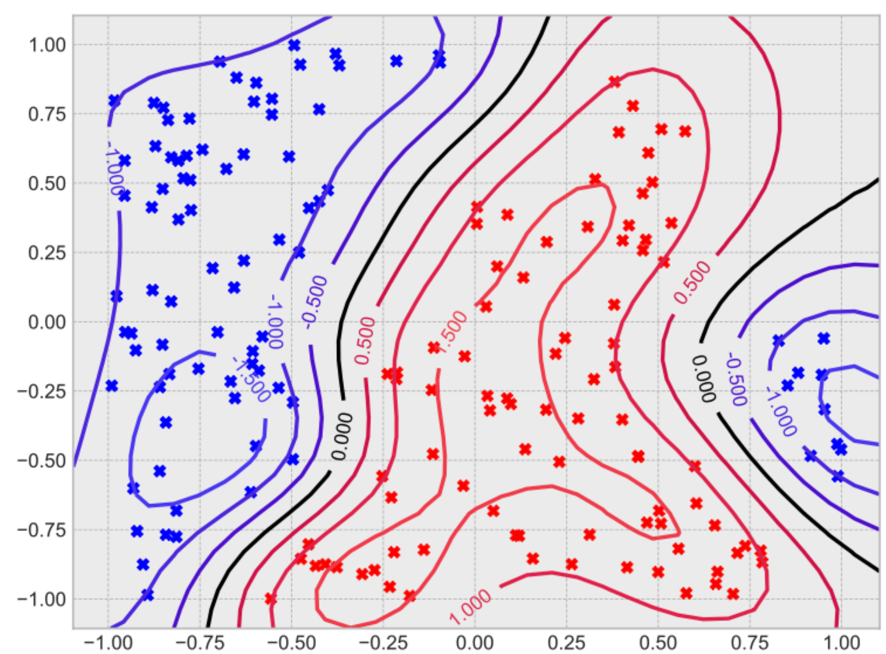
Source data.



Quadratic SVM.



RBF SVM ($\sigma = 1$).



RBF SVM ($\sigma = 0.1$).

Next Time

- Decision Trees
- Random Forests