Machine Learning

Section 11: Support Vector Machines

Stefan Harmeling

15./17./22. November 2021

What we have seen so far?

Sections:

1. Introduction
2. Plausible reasoning and Bayes Rule
3. From Logic to Probabilities
4. Bayesian networks
5. Continuous Probabilities
6. The Gaussian distribution
7. More on distributions, models, MAP, ML
8. Linear Regression
9. Matrix Differential Calculus
10. Model selection

And now for something completely different.

Classification problems

Classification problem

- given patterns $x_1, \ldots, x_n \in \mathbb{R}^M$
- and class labels $y_1, \ldots, y_n \in \{+1, -1\}$
- find a decision function $f(x)$ that predicts the label y_* of a new pattern x_*.

Notions:

- Classification problems are supervised learning problems, since the labels are given.
- Separable case: there is a gap between the two classes, so there is a perfect decision function.
- Non-separable case: the classes are overlapping, so the decision function will make some mistakes.
- Linear case: the true decision function is linear.
- Nonlinear case: the true decision function is nonlinear.

Classifiers

- Methods/algorithms that solve classification problems.

Notions

- Linear classifiers learn a *linear* decision function from data.
- Nonlinear classifiers learn a *nonlinear* decision function from data.

This section on SVM has three parts:

1. Linearly separable case (TODAY)
2. Linearly non-separable case
3. Nonlinear case

Part 1: The linearly separable case

Today

Linearly separable classification problem

- we assume that the patterns $x_1, \ldots, x_n$ are linearly separable in two classes $+1, -1$ as described by their labels $y_1, \ldots, y_n$

Preview:

Linear Support Vector Machine (SVM)

- a supervised learning method for the classification problem
- cuts the space of patterns into two parts depending on the labels by choosing the unique *max margin* hyperplane
- achieves this by solving a *constrained optimization problem*

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 && \textit{maximize the margin} \\ \text{s.t.} \quad & y_i\,(<w, x_i> + b) \geq 1 \text{ for all } i && \textit{correctly classify all data} \end{aligned}$$

- Question: How can we derive this?

How maximizing the margin is a constrained optimization problem

Decision functions based on hyperplanes

Hyperplane

- A *hyperplane* is given by a pair (w, b), where the hyperplane consists of all points x, s.t.

$$<w, x> + b = w^T x + b = 0 \text{ where } w \in \mathbb{R}^d, b \in \mathbb{R}$$

- A hyperplane is called *separating* for data $x_1, \ldots, x_n$ with labels $y_1, \ldots, y_n$ if for all i

$$<w, x_i> + b > 0 \qquad \text{for "positive" } x_i \text{, i.e. all } i \text{ with } y_i = +1$$
$$<w, x_i> + b < 0 \qquad \text{for "negative" } x_i \text{, i.e. all } i \text{ with } y_i = -1$$

or written more conveniently (this is reason for using +1/-1 as labels):

$$y_i \left(<w, x_i> + b\right) > 0 \qquad \text{for all } i$$

Decision function (aka rule) based on a hyperplane:

$$f(x) = \operatorname{sgn}(<w, x> + b) = \begin{cases} +1 & \text{if } <w, x> + b \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Canonical separating hyperplanes

Note

- if (w, b) is a separating hyperplane for some data, then $(c\,w, c\,b)$ describes the *same* separating hyperplane for any scalar $c > 0$, i.e.

$$y_i\,(<w, x_i> + b) > 0 \iff y_i\,(<c\,w, x_i> + c\,b) > 0$$

- How should we choose the constant?

Canonical separating hyperplane

- The expression $y_i\,(<w, x_i> + b)$ is called the *functional margin* of x_i.
- A separating hyperplane is called *canonical* if the smallest functional margin is one, i.e.

$$\min_i y_i\,(<w, x_i> + b) = 1$$

- Any separating hyperplane can be made canonical by rescaling.
- Making a hyperplane canonical isn't changing its decision function.

From the functional to the geometrical margin

Geometrical margin

- The *geometrical margin* can be obtained from the functional margin by scaling with $1/\|w\|$,

$$\frac{1}{\|w\|} y_i \left(<w, x_i> + b\right) = y_i \left(<\frac{w}{\|w\|}, x_i> + \frac{b}{\|w\|}\right)$$

- The geometrical margin measures the distance of a data point x_i to the separating hyperplane, since the inner product of x_i with a normalized vector $w/\|w\|$ measures the length of the projection of x_i onto w from the origin.

Note

- The geometrical margins allow us to choose between hyperplanes that have different decision functions.

Maximizing the margin

Margin

The *margin* of a canonical separating hyperplane (w, b) is the minimal geometrical margin, i.e.

$$\min_i \frac{1}{\|w\|} y_i \left(<w, x_i> + b\right) = \frac{1}{\|w\|} \min_i y_i \left(<w, x_i> + b\right) = \frac{1}{\|w\|}$$

Max margin classifier

- Maximize the margin by minimizing $\|w\|$.
- Ensure that the hyperplane (w, b) is separating the data, by fulfilling the constraints $y_i \left(<w, x_i> + b\right) \geq 1$ for all i
- Write this as a constrained optimization problem:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 && \textit{maximize the margin} \\ \text{s.t.} \quad & y_i \left(<w, x_i> + b\right) \geq 1 \text{ for all } i && \textit{correctly classify all data} \end{aligned}$$

Q: Can we replace 1 by some other number?

Constrained optimization problems

Primal problem

Maximize the margin while ensuring separation:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i\,(<w, x_i> + b) \geq 1 \text{ for all } i \end{aligned}$$

How to solve this?

- plug it directly into some optimization software
- apply the method of Lagrange multiplier to derive the dual problem
- then solve the dual problem (which is sometimes easier)

Lagrange duality

- we are closely following Sec. 5 in Andrew Ng's lecture notes for his Stanford course CS229, which is also a good read for the whole lecture's topic
- two related excellent books of Stephen Boyd
  - "Convex optimization", 2004, `http://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf`
  - "Applied linear algebra", 2018 `https://web.stanford.edu/~boyd/vmls/vmls.pdf`

The method of Lagrange multiplier

Constrained optimization problem

$\min_w$	$f(w)$	objective function of vector w
s.t.	$h_i(w) = 0$ for $i = 1, \ldots, l$	equality constraints

How to solve such a problem?

- define Lagrangian (function)

$$\mathcal{L}(w, \beta) = f(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

where $\beta = [\beta_1, \ldots, \beta_l]$ is the vector of *Lagrange multipliers*

- calculate partial derivatives for w and β

$$\frac{\partial \mathcal{L}}{\partial w_i} = 0 \qquad \frac{\partial \mathcal{L}}{\partial \beta_i} = 0$$

- solve for w and β

Why we set the gradient of the Lagrangian to zero

Constrained optimization problem

On this slide we optimize wrt. (x, y) instead of w.

$$\min_{x,y} \quad f(x,y) \qquad \text{s.t.} \quad g(x,y) = 0$$

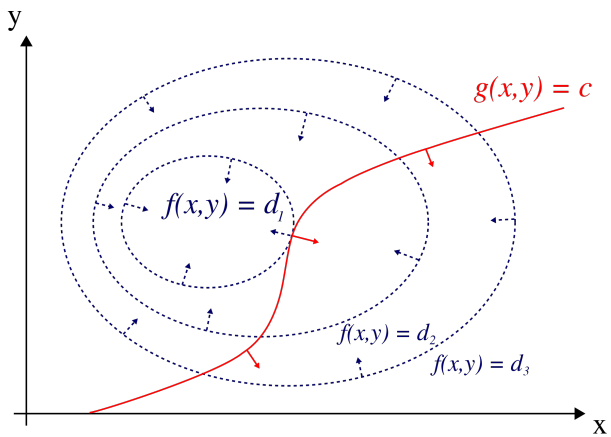


Fig. from https://en.wikipedia.org/wiki/Lagrange_multiplier

Lagrangian

$$\mathcal{L}(x, y, \lambda) = f(x,y) - \lambda g(x,y)$$

Setting gradients of $\mathcal{L}$ wrt. x, y, λ to zero implies

$$\nabla_{x,y} f = \lambda \nabla_{x,y} g \qquad g(x,y) = 0$$

More constaints: inequalities and equalities

Primal optimization problem

$\min_w$	$f(w)$	objective function of vector w
s.t.	$g_i(w) \leq 0$ for $i = 1, \ldots, k$	inequality constraints
	$h_i(w) = 0$ for $i = 1, \ldots, l$	equality constraints

Generalized Lagrangian

$$\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

- α and β are vectors Lagrange multiplier, also assume $\alpha_i \geq 0$
- w is called *primal* variable, α, β *dual* variables
- define optima wrt to w and wrt to α and β

$$\theta_{\mathcal{P}}(w) = \max_{\alpha, \beta : \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta) \qquad \mathcal{P} \text{ is primal}$$

$$\theta_{\mathcal{D}}(\alpha, \beta) = \min_{w} \mathcal{L}(w, \alpha, \beta) \qquad \mathcal{D} \text{ is dual}$$

$$\theta_{\mathcal{P}}(w) = \max_{\alpha,\beta:\alpha_i \geq 0} \mathcal{L}(w,\alpha,\beta) \qquad \mathcal{P} \text{ is primal}$$

$$\theta_{\mathcal{D}}(\alpha,\beta) = \min_{w} \mathcal{L}(w,\alpha,\beta) \qquad \mathcal{D} \text{ is dual}$$

Question

What is the relationship between primal and dual?

To understand thisrelationship let's look at the *unconstrained* problem

How to get an unconstrained problem

Consider:

$$\theta_{\mathcal{P}}(w) = \max_{\alpha,\beta:\alpha_i \geq 0} \mathcal{L}(w,\alpha,\beta) = \max_{\alpha,\beta:\alpha_i \geq 0} f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

$$= \begin{cases} f(w) & \text{if } w \text{ satifies primal constraints} \\ \infty & \text{otherwise} \end{cases}$$

- note that if $h_i(w) = 0$ the last sum is zero
- note that if $g_i(w) \leq 0$ the second sum is ≤ 0 since $\alpha_i \geq 0$

Unconstrained problem

- primal value:

$$p^* := \min_w \theta_{\mathcal{P}}(w) = \min_w \max_{\alpha,\beta:\alpha_i \geq 0} \mathcal{L}(w,\alpha,\beta)$$

- thus we can minimize $\theta_{\mathcal{P}}(w)$ wrt. w without any constraints instead of the initial constrained problem
- however, it might be difficult to get an expression for $\theta_{\mathcal{P}}$
- alternative: the dual problem

The dual problem

Consider:

$$\theta_{\mathcal{P}}(w) = \max_{\alpha,\beta:\alpha_i \geq 0} \mathcal{L}(w,\alpha,\beta) \qquad \mathcal{P} \text{ is primal}$$

$$\theta_{\mathcal{D}}(\alpha,\beta) = \min_{w} \mathcal{L}(w,\alpha,\beta) \qquad \mathcal{D} \text{ is dual}$$

Dual value

$$d^* := \max_{\alpha,\beta:\alpha_i \geq 0} \theta_{\mathcal{D}}(\alpha,\beta) = \max_{\alpha,\beta:\alpha_i \geq 0} \min_{w} \mathcal{L}(w,\alpha,\beta)$$

- also called the dual problem
- same as the primal problem but now min und max interchanged
- sometimes the constraints $\alpha_i \geq 0$ are “easier” than the initial constraints on w in the primal problem

Relationship between primal and dual

In general:

$$d^* = \max_{\alpha,\beta:\alpha_i \geq 0} \theta_{\mathcal{D}}(\alpha,\beta) \leq \min_{w} \theta_{\mathcal{P}}(w) = p^*$$

- difference $p^* - d^*$ is called *duality gap*

For convex f, g_i and affine h_i:

$$d^* = p^*$$

- the problem is *feasible*, if there exists w^*, α^*, β^* as solutions for the primal and dual
- then the Karush-Kuhn-Tucker (KKT) conditions hold, e.g.

$$\alpha_i^* g_i(w^*) = 0$$

- there are more KKT conditions...

Constrained optimization — summary

Primal problem:

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \leq 0 \text{ for } i = 1, \ldots, k \\ & h_i(w) = 0 \text{ for } i = 1, \ldots, l \end{aligned}$$

Unconstrained primal problem:

$$\min_w \quad \theta_{\mathcal{P}}(w)$$

Dual problem:

$$\begin{aligned} \max_{\alpha,\beta:\alpha_i \geq 0} \quad & \theta_{\mathcal{D}}(\alpha, \beta) \\ \text{s.t.} \quad & \alpha_i \geq 0 \end{aligned}$$

Pick the one which is simpler to solve!

Back to SVMs...

Deriving the dual problem for the SVM

- Lagrangian with dual variables $\alpha_i \geq 0$

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i (y_i(<w, x_i> + b) - 1)$$

- Eliminate w and b by plugging in the zeroed derivatives for the saddle point

$$\frac{\partial}{\partial w}\mathcal{L}(w, b, \alpha) = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0, \qquad \frac{\partial}{\partial b}\mathcal{L}(w, b, \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0$$

to get the dual problem...

Dual problem

Instead of minimizing in w and b, here we maximize wrt α:

$$\max_\alpha \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j <x_i, x_j>$$

$$\text{s.t.} \quad \alpha_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

Brief summary (separable case)

Primal problem

Maximize the margin while ensuring separation:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i \left(<w, x_i> + b\right) \geq 1 \text{ for all } i \end{aligned}$$

Dual problem

Instead of minimizing in w and b, here we maximize wrt α:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \, y_i y_j \, <x_i, x_j> \\ \text{s.t.} \quad & \alpha_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{n} \alpha_i \, y_i = 0 \end{aligned}$$

Linear SVM algorithm (separable case)

- given training data $(x_1, y_1), \ldots, (x_n, y_n)$
- find α that solves the dual problem

$$\max_\alpha \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \, y_i y_j \, <x_i, x_j>$$

$$\text{s.t.} \quad \alpha_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{n} \alpha_i \, y_i = 0$$

- use α for the decision function

$$f(x) = \operatorname{sgn}(<w, x> + b) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i \, y_i \, <x_i, x> + b \right)$$

where $w = \sum_{i=1}^{n} \alpha_i \, y_i \, x_i$ follows from $\partial / \partial w L = 0$ and b is calculated from the complementary KKT conditions (see next slides).

Support vectors (separable case)

The solution α of the dual problems fulfills:

Complementary KKT condition:

$$\alpha_i(y_i(<w, x_i>+b) - 1) = 0 \text{ for all } i$$

Case distinction:

- $\alpha_i > 0$: then $y_i(<w, x_i>+b) = 1$ and x_i must lie exactly on the margin. x_i is a support vector. Call the set of the indices of support vectors $\mathcal{S}$.
- $\alpha_i = 0$: then $y_i(<w, x_i>+b) > 1$ (note that it can not be < 1 since otherwise α_i could grow arbitrarily), and x_i is not a support vector.

How to calculate w and b given α?

w:

From the derivative of the Lagrangrian wrt. w, we get:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i = \sum_{i \in \mathcal{S}} \alpha_i y_i x_i$$

b:

Transforming $\alpha_i(y_i(<w, x_i> + b) - 1) = 0$ for $\alpha_i > 0$, i.e. for $i \in \mathcal{S}$ we get:

$$b_i = \frac{1}{y_i} - <w, x_i> = y_i - <w, x_i>$$

All b_i are approximately the same, define b as their average:

$$b = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} y_i - <w, x_i>$$

How to implement SVMs?

Implementing SVMs (separable case), (1)

Dual of linear SVM

$$\max_\alpha \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j <x_i, x_j>$$

$$\text{s.t.} \quad \alpha \geq 0 \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

Quadratic programming (QP, quadprog)

$$\min_x \quad \frac{1}{2} x^T Q x + c^T x$$

$$\text{s.t.} \quad Ax \leq b$$

$$A_{\text{eq}} x = b_{\text{eq}}$$

$$x_{\text{l}} \leq x \leq x_{\text{u}}$$

Implementing SVMs (separable case), (2)

Dual of linear SVM

$$\max_\alpha \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j <x_i, x_j>$$

$$\text{s.t.} \quad \alpha \geq 0 \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

Dual of linear SVM rewritten for QP

$$\min_\alpha \quad \frac{1}{2} \alpha^T Q \alpha - 1_n^T \alpha$$

$$\text{s.t.} \quad y^T \alpha = 0$$

$$0 \leq \alpha$$

- with $Q = \text{Diag}(y) X X^T \text{Diag}(y)$
- with 1_n being n dimensional ones vector

Implementing SVMs (separable case), (3)

Dual of linear SVM rewritten for QP

$$\begin{aligned} \min_\alpha \quad & \frac{1}{2}\alpha^T Q\alpha - 1_n^T\alpha \\ \text{s.t.} \quad & y^T\alpha = 0 \\ & 0 \le \alpha \end{aligned}$$

Quadratic programming (QP, quadprog)

$$\begin{aligned} \min_x \quad & \frac{1}{2}x^T Qx + c^T x \\ \text{s.t.} \quad & Ax \le b \\ & A_{\text{eq}}x = b_{\text{eq}} \\ & x_{\text{l}} \le x \le x_{\text{u}} \end{aligned}$$

- $x = \alpha$, $c = -1_n$, $A = b = [\,]$, $x_{\text{l}} = 0$, $x_{\text{u}} = \infty$, $A_{\text{eq}} = y^T$, $b_{\text{eq}} = 0$

Implementing QPs (1)

Quadratic programming (QP, quadprog)

$$\begin{aligned} \min_x \quad & \frac{1}{2} x^T Q x + c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & A_{\text{eq}} x = b_{\text{eq}} \\ & x_{\text{l}} \leq x \leq x_{\text{u}} \end{aligned}$$

Interior point nonlinear optimization

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & g_l \leq g(x) \leq g_u \\ & x_{\text{l}} \leq x \leq x_{\text{u}} \end{aligned}$$

- implement objective and constraints $f(x)$, $g(x)$
- implement gradients $\nabla_x f(x)$, $\nabla_x g(x)$
- implement Hessian of Lagrangian $\nabla_x^2 \mathcal{L}(x)$

Implementing QPs (2)

Objective function and constraints

$$f(x) = \frac{1}{2} x^T Q x + c^T x \qquad \nabla_x f(x) = Qx + c$$

$$g(x) = \begin{bmatrix} A \\ A_{\text{eq}} \end{bmatrix} x \qquad \nabla_x g(x) = \begin{bmatrix} A \\ A_{\text{eq}} \end{bmatrix}$$

$$g_l = \begin{bmatrix} -\infty_m \\ b_{\text{eq}} \end{bmatrix} \qquad g_u = \begin{bmatrix} b \\ b_{\text{eq}} \end{bmatrix}$$

- m is number of rows of A
- ∞_m is m dimensional ones vector times ∞

Hessian of Langrangian

$$\nabla_x^2 \mathcal{L}(x) = Q$$

- constraints are linear, so their Hessian is zero
- objective is quadratic, so its Hessian is just Q

Implementing SVMs (separable case) - summary

Primal problem

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i\,(<w, x_i> + b) \geq 1 \text{ for all } i \end{aligned}$$

Dual problem

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j <x_i, x_j> \\ \text{s.t.} \quad & \alpha \geq 0 \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned}$$

- formulate the optimization problem as a QUADPROG
- call QUADPROG library function
- if no QUADPROG available, look for general optimizer and implement QUADPROG yourself
- if no general optimizer, (ups) ..., implementing a general optimizer is interesting as well, but more difficult...

So far (done):

- Part 1: The linearly separable case

Next:

- Interlude: The method of Lagrange/KKT multiplier in a nutshell
- Part 2: The linearly non-separable case
- Part 3: The nonlinear case

More Literature:

- Mohammed J. Zaki, Wagner Meira, Jr., Data Mining and Machine Learning: Fundamental Concepts and Algorithms, Cambridge University Press, `https://dataminingbook.info/book_html/` (contains detailed derivation of b which is often missing in many presentations)

Method of Lagrange multiplier/KKT multiplier in a nutshell

(this time without trying to explain why it works)

Nonlinear optimization with equality constraints

Goal: solve this optimization problem

$\min_w$	$f(w)$	objective function of vector w
	$h_i(w) = 0$ for $i = 1, \ldots, l$	equality constraints

Question: how we do that?

Method of Lagrange multiplier

$$\begin{array}{lll} \min_w & f(w) & \text{objective function of vector } w \\ & h_i(w) = 0 \text{ for } i = 1, \ldots, l & \text{equality constraints} \end{array}$$

- Option 1: plug it directly into a solver
- Option 2: "method of Lagrange multiplier"
  - transform it into an *unconstrained* problem by defining the Lagrangian function

$$\mathcal{L}(w, \beta) = f(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

  - write out the dual problem

$$\begin{array}{lll} \max_\beta & \min_w \mathcal{L}(w, \beta) & \text{objective function of vector } \beta \end{array}$$

    where we can ideally eliminate w using $\nabla_w \mathcal{L}(w, \beta) = 0$
  - then plug it into a solver

Nonlinear optimization with general constraints

Goal: solve this optimization problem

$\min_w$	$f(w)$	objective function of vector w
s.t.	$g_i(w) \leq 0$ for $i = 1, \ldots, k$	inequality constraints
	$h_i(w) = 0$ for $i = 1, \ldots, l$	equality constraints

Question: how we do that?

Method of KKT multiplier

$\min_w$	$f(w)$	objective function of vector w
s.t.	$g_i(w) \leq 0$ for $i = 1, \ldots, k$	inequality constraints
	$h_i(w) = 0$ for $i = 1, \ldots, l$	equality constraints

- Option 1: plug it directly into a solver
- Option 2: "method of KKT multiplier"
  - transform it into an *unconstrained* problem by defining the generalized Lagrangian function

$$\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

  - write out the dual problem

$$\max_{\alpha,\beta} \quad \min_w \mathcal{L}(w, \alpha, \beta) \qquad \text{objective function of vectors } \alpha, \beta$$

    where we can ideally eliminate w using $\nabla_w \mathcal{L}(w, \alpha, \beta) = 0$
  - plug it into a solver

Two options for the separable case (last time)

Given $x_1, \ldots, x_n \in \mathbb{R}^d$ and $y_1, \ldots, y_n \in \{-1, +1\}$.

Primal problem

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i(<w, x_i> + b) \geq 1 \end{aligned}$$

Dual problem

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \, y_i y_j \, <x_i, x_j> \\ \text{s.t.} \quad & \alpha_i \geq 0 \\ & \sum_{i=1}^{n} \alpha_i \, y_i = 0 \end{aligned}$$

Both can be implemented using `quadprog`.
Let's implement the primal as well.

Convexity

Definition 11.1

A function $f : \mathbb{R} \to \mathbb{R}$ *is convex if for all* $x_1, x_2 \in \mathbb{R}$ *and* $0 \leq \gamma \leq 1$ *we have*

$$f(\gamma x_1 + (1-\gamma)x_2) \leq \gamma f(x_1) + (1-\gamma) f(x_2)$$

- A function f is concave if $-f$ is convex.
- Convexity is similar to linearity but replaces equality with an inequality.
- A local minimum of a convex function is global.
- Jensen's inequality holds for convex functions f:

$$\mathrm{E}\, f(X) \geq f(\mathrm{E}\, X)$$

- The area above a convex function is a convex set (i.e. every line between points of the set is inside the set).

Karush-Kuhn-Tucker (KKT) conditions

$$\begin{aligned} \min_w \quad & f(w) && \text{objective function of vector } w \\ \text{s.t.} \quad & g_i(w) \leq 0 \text{ for } i = 1, \ldots, k && \text{inequality constraints} \\ & h_i(w) = 0 \text{ for } i = 1, \ldots, l && \text{equality constraints} \end{aligned}$$

$$\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w) \qquad \text{generalized Lagrangian}$$

Assume f, all g_i are convex, and all h_i is affine (i.e. “shifted linear”).

Theorem 11.2

A point (w^*, α^*, β^*) is optimal if and only if it fulfills the KKT conditions:

$$\begin{aligned} \nabla_w \mathcal{L}(w^*, \alpha^*, \beta^*) = 0 && \textit{stationary} \\ g_i(w^*) \leq 0 \textit{ for all } i && \textit{primal admissibility} \\ h_i(w^*) = 0 \textit{ for all } i && \textit{primal admissibility} \\ \alpha_i \geq 0 \textit{ for all } i && \textit{dual admissibility} \\ \alpha_i g_i(w^*) = 0 \textit{ for all } i && \textit{complementary slackness} \end{aligned}$$

Part 2: The linearly unseparable case

Separable vs non-separable case

Separable case

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}||w||^2 \\ \text{s.t.} \quad & y_i(<w, x_i> + b) \geq 1 \text{ for all } i \end{aligned}$$

Non-separable case

- Some constraints $y_i(<w, x_i> + b) \geq 1$ are always violated.
- So there is no feasible solution (one that fulfills *all* constraints).
- How can we *relax* the optimization problem?

Solution:

- Relax the problem by introducing slack variables $\xi_i \geq 0$, s.t.

$$y_i(<w, x_i> + b) \geq 1 - \xi_i \qquad \text{for all } i$$

- Keep the slack variables small, i.e. minimize $\sum_i \xi_i$
- A new hyperparameter C that has to be tuned, that trades off $0.5||w||^2$ and $\sum_i \xi_i$.

Hard margin vs soft margin

Primal problem for hard margin (aka separable case)

Assuming a separating hyperplane exist:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i(<w, x_i> + b) \geq 1 \text{ for all } i \end{aligned}$$

Primal problem for soft margin (aka non-separable case)

Assume data points that violate the separating hyperplane:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i(<w, x_i> + b) \geq 1 - \xi_i \text{ for all } i \\ & \xi_i \geq 0 \text{ for all } i \end{aligned}$$

- new hyperparameter C that has to be tuned (e.g. cross validation)

Deriving the dual of the soft-margin formulation

Primal problem for soft margin (aka non-separable case)

Assume data points that violate the separating hyperplane:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i(<w, x_i> + b) \geq 1 - \xi_i \text{ for all } i \\ & \xi_i \geq 0 \text{ for all } i \end{aligned}$$

Lagrangian with dual variables $\alpha_i \geq 0$ and $\beta_i \geq 0$

$$\mathcal{L}(w, b, \xi, \alpha, \beta) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(<w, x_i> + b) - 1 + \xi_i) - \sum_{i=1}^{n}\beta_i\xi_i$$

Deriving the dual of the soft-margin formulation

Lagrangian with dual variables $\alpha_i \geq 0$ and $\beta_i \geq 0$

$$\mathcal{L}(w, b, \xi, \alpha, \beta) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(<w, x_i>+b) - 1 + \xi_i) - \sum_{i=1}^{n}\beta_i\xi_i$$

$$= \frac{1}{2}<w, w> - \sum_{i=1}^{n}\alpha_i(y_i(<w, x_i>+b) - 1) + \sum_{i=1}^{n}(C - \alpha_i - \beta_i)\xi_i$$

$$= \frac{1}{2}<w, w> - <w, \sum_{i=1}^{n}\alpha_i y_i x_i> + \sum_{i=1}^{n}\alpha_i - b\sum_{i=1}^{n}\alpha_i y_i + \sum_{i=1}^{n}(C - \alpha_i - \beta_i)\xi_i$$

Calculate the derivatives with respect to the primal variables:

$$\frac{\partial}{\partial w}\mathcal{L}(w, b, \xi, \alpha, \beta) = w - \sum_{i=1}^{n}\alpha_i\, y_i\, x_i = 0$$

$$\frac{\partial}{\partial b}\mathcal{L}(w, b, \xi, \alpha, \beta) = -\sum_{i=1}^{n}\alpha_i\, y_i = 0$$

$$\frac{\partial}{\partial \xi_i}\mathcal{L}(w, b, \xi, \alpha, \beta) = C - \alpha_i - \beta_i = 0$$

Plug these equalities into the Lagrangian!

$$\mathcal{L}(w, b, \xi, \alpha, \beta) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} <w, w> = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j <x_i, x_j>$$

- we can eliminate the dual variable β_i by combining $\alpha_i \geq 0$ and $\beta_i \geq 0$ with $C - \alpha_i = \beta_i$ to get $C \geq \alpha_i \geq 0$

Thus we arrived at the dual problem for the non-separable case:

Dual problem (non-separable case)

Instead of minimizing in w, b and ξ, we maximize wrt α:

$$\max_\alpha \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \, y_i y_j <x_i, x_j>$$

$$\text{s.t.} \quad C \geq \alpha_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{n} \alpha_i \, y_i = 0$$

Brief summary for the soft margin

Assume possible outliers that violate the separating hyperplane.

Primal problem for soft margin

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i(<w, x_i>+b) \geq 1-\xi_i \text{ for all } i \\ & \xi_i \geq 0 \text{ for all } i \end{aligned}$$

Dual problem for soft margin

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j\, y_i y_j\, <x_i, x_j> \\ \text{s.t.} \quad & C \geq \alpha_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{n}\alpha_i\, y_i = 0 \end{aligned}$$

Q: can you spot the difference in the dual for the hard margin?

How to calculate w and b given α?

(non-separable case)

Support vectors (non-separable case)

The solution α of the dual problems fulfills:

Complementary KKT conditions:

$$\alpha_i(y_i(<w, x_i>+b) - 1 + \xi_i) = 0 \qquad \text{for all } i$$
$$\beta_i\xi_i = (C - \alpha_i)\xi_i = 0 \qquad \text{for all } i$$

where we removed β_i using $C - \alpha_i - \beta_i = 0$.

Case distinction:

- $0 < \alpha_i < C$: then $(C - \alpha_i)\xi_i = 0$ implies $\xi_i = 0$ and the other condition implies $y_i(<w, x_i>+b) = 1$, and x_i lies on the margin, which can use to estimate the offset b. These x_i are called support vectors.
- $\alpha_i = 0$: then $(C - \alpha_i)\xi_i = 0$ implies also $\xi_i = 0$ and $y_i(<w, x_i>+b) > 1$ and x_i must lie outside the margin.
- $\alpha_i = C$: then $y_i(<w, x_i>+b) = 1 - \xi_i$ and x_i must lie inside the margin. The point x_i is also called support vector.

How to calculate w and b given α?

w:
From the derivative of the Lagrangrian wrt. w, we get:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i = \sum_{i:\alpha_i>0} \alpha_i y_i x_i$$

b:
Transforming $\alpha_i(y_i(<w, x_i>+b) - 1 + \xi_i) = 0$ for $C > \alpha_i > 0$ we get:

$$b_i = \frac{1}{y_i} - <w, x_i> = y_i - <w, x_i>$$

All b_i are approximately the same, define b as their average:

$$b = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} y_i - <w, x_i>$$

Linear SVM algorithm (dual solution)

- given training data $(x_1, y_1), \ldots, (x_n, y_n)$
- find α that solves the dual problem

$$\max_\alpha \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \, y_i y_j \, <x_i, x_j>$$

$$\text{s.t.} \quad C \geq \alpha_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{n} \alpha_i \, y_i = 0$$

- use α for the decision function

$$f(x) = \operatorname{sgn}(<w, x> + b) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i \, y_i \, <x_i, x> + b \right)$$

where $w = \sum_{i=1}^{n} \alpha_i \, y_i \, x_i$ and b is calculated from the KKT conditions

Solve the primal or the dual?

Given $x_1, \ldots, x_n \in \mathbb{R}^d$ and $y_1, \ldots, y_n \in \{-1, +1\}$.

Primal problem

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(<w, x_i> + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned}$$

- $d + n + 1$ unknown variables
- $2n$ constraints
- use when $d < n$

Dual problem

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j <x_i, x_j> \\ \text{s.t.} \quad & C \geq \alpha_i \geq 0 \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned}$$

- n unknown variables
- requires $n \times n$ matrix with entries $<x_i, x_j>$
- $2n$ simple box constraints
- use when $d \gg n$
- (use when solving the nonlinear setup, later. . .)

Linear SVM formulation (non-separable case)

Given $x_1, \ldots, x_n \in \mathbb{R}^d$ and $y_1, \ldots, y_n \in \{-1, +1\}$.

Primal problem

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(<w, x_i> + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned}$$

Dual problem

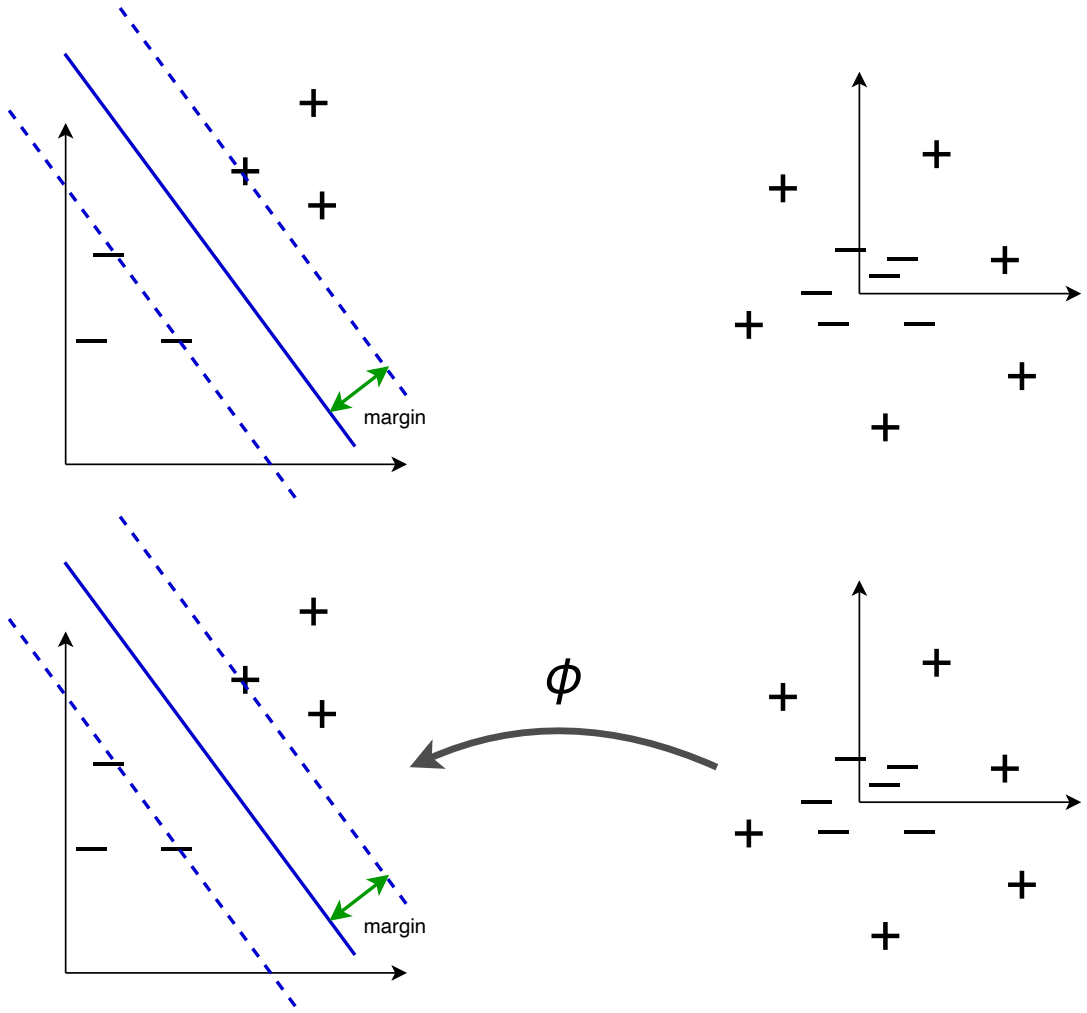
$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j \, y_i y_j \, <x_i, x_j> \\ \text{s.t.} \quad & C \geq \alpha_i \geq 0 \\ & \sum_{i=1}^{n} \alpha_i \, y_i = 0 \end{aligned}$$

Question

- How do we get a nonlinear SVM?

Part 3: The nonlinear case

Linear classification vs nonlinear classification



Idea: map to a feature space where the classes are linearly separable.

Example mapping

see Jupyter notebook

The kernel trick

- see also Schölkopf, Mika, Burgers, Knirsch, Müller, Rätsch, Smola, “Input Space vs. Feature Space in Kernel-Based Methods”, 1999

Classification problem

- given patterns $x_1, \ldots, x_n \in \mathcal{X}$ (e.g. $\mathcal{X} = \mathbb{R}^M$)
- and class labels $y_1, \ldots, y_n \in \mathcal{Y} = \{+1, -1\}$
- find a decision function $f : \mathcal{X} \to \mathcal{Y}$ that predicts the label y_* of a new pattern x_*.

Notions:

- Linear case: the true decision function is linear.
- Nonlinear case: the true decision function is nonlinear.

From input space to feature space

Not linearly separable:

- e.g. class 1 close at the origin, class -1 further away.
- idea: map the data to new features, e.g.

input spacefeature space

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \longmapsto \phi(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 + x_2^2 \end{bmatrix}$$

- the polar coordinates is an example of a *feature space*, where the classes are linearly separable

Notational warning:

- on this and the next slides we use x_i to denote coordinates of x
- in most of the remaining slides x_i is again the ith data point

Feature maps and inner products

Feature map (another example):

$$\begin{array}{cccc} & \text{input space} & & \text{feature space} \\ \phi: & \mathbb{R}^2 & \longrightarrow & \mathbb{R}^3 \\ & \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} & \longmapsto & \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\, x_1 x_2 \end{bmatrix}. \end{array}$$

Linear SVM looks at the data through dot products:

$$\max_\alpha \quad \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j <x_i, x_j>$$

$$\text{s.t.} \quad C \geq \alpha \geq 0 \text{ and } \sum_{i=1}^n \alpha_i y_i = 0$$

Dot product is essential to compare data points:

e.g. distance can be written with dot products

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

Feature maps induce kernel functions

Dot product in feature space:

$$
\begin{aligned}
\langle \phi(x), \phi(x') \rangle &= \phi(x)^\mathsf{T} \phi(x') \\
&= \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\, x_1 x_2 \end{bmatrix}^\mathsf{T} \begin{bmatrix} x'^2_1 \\ x'^2_2 \\ \sqrt{2}\, x'_1 x'_2 \end{bmatrix} \\
&= x_1^2 x'^2_1 + x_2^2 x'^2_2 + 2 x_1 x_2 x'_1 x'_2 \\
&= (x^\mathsf{T} x')^2 =: k(x, x').
\end{aligned}
$$

- dot product in feature space is a non-linear function k in input space, called *kernel function*
- can be calculated without explicitly mapping to the feature space via ϕ
- ϕ induces a kernel function

Question:

Can we avoid defining ϕ and directly specify k?

Kernel functions

Answer:

Yes! E.g. $k(x, x') = (x^\mathsf{T} x')^2$ can be generalized:

$$k(x, x') = (x^\mathsf{T} x')^p$$

- called *homogeneous polynomial kernel*
- one can show: $k(x, x') = \phi(x)^\mathsf{T} \phi(x')$ with

$$\begin{array}{ccccc} & \text{input space} & & \text{feature space} \\ \phi: & \mathbb{R}^d & \longrightarrow & \mathbb{R}^D \\ & \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} & \longmapsto & \begin{bmatrix} x_1^p \\ x_1^{p-1} x_2 \\ \vdots \end{bmatrix} \end{array}$$

- i.e. x is mapped to all the monomials of degree p
- note that $D \gg d$
- kernel function calculates inner product in $\mathbb{R}^D$ without calculating all monomials which might be computationally prohibitive

Common kernel functions (1)

from Murphy Chapter 14

- Polynomial kernel

$$k(x, x') = (x^T x' + b)^p$$

called *homogenous* for $b = 0$ (all monomials of degree p, *inhomogenous* otherwise (all monomials up to degree p)

- Linear kernel

$$k(x, x') = x^T x'$$

special case of the polynomial kernel

Common kernel functions (2)

from Murphy Chapter 14

- Gaussian kernel, aka RBF kernel, aka squared exponenetial kernel

$$k(x, x') = \exp\left(-\frac{(x - x')^T (x - x')}{2\sigma^2}\right)$$

  where σ^2 is called the bandwidth
- Why Gaussians?
- The Gaussian kernel does measure similarity, e.g. $1 = k(x, x) \geq k(x, x')$, for other x'.
- The corresponding feature space is ∞-dimensional...

Common kernel functions (3)

from Murphy Chapter 14

Measuring similarity between documents

- Summarize documents by bag of word representation, i.e. for document i let x_{ij} be the number of times word j appears in document i.
- Cosine similarity

$$k(x, x') = \frac{x^T x'}{\|x\|_2 \|x'\|_2}$$

- Note:
  - normalized inner product, i.e. the length doesn't matter
  - measures the cosine of the angle between two documents
  - $x_{ij} \geq 0$ so $0 \leq k(x, x') \leq 1$
  - in practice this should be more sophisticated: use tf-idf (term frequency inverse document frequency) representation

Common kernel functions (4)

from Murphy Chapter 14.2.6

Comparing two strings of variable lengths:

- how similar are

$$x = \texttt{LKSDJFLJWELELVISKJNEROIUSGOGFKJ}$$
$$x' = \texttt{OREIUTKELVISGNKJNRGKJNREKJ}$$

- s is a substring of x, if $x = usv$ for possibly zero-length strings u and v
- let $\phi_s(x)$ be the number of times that s appears in x
- count the number of substrings have in common, i.e.

$$k(x, x') = \sum_{s \in \mathcal{A}^*} w_s \phi_s(x) \phi_s(x')$$

  where $w_s \geq 0$ and $\mathcal{A}^*$ is the set of all strings
- surprisingly this can be calculated in $O(|x| + |x'|)$ using suffix trees
- $\longrightarrow$ Bioinformatics ($\mathcal{A} = \{ACTG\}$)

The famous kernel trick

Goal:

- create a nonlinear version of an existing linear algorithm

Requirement:

- the linear algorithm must only calculate dot products of the data points

Kernelization:

- replace all dot products by a kernel function

Examples:

- kernel PCA, kernel LDA, kernel CCA, kernel FDA, kernel ICA, ...

Kernelizing the Linear SVM (1)

Linear problem:

$$\max_\alpha \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j <x_i, x_j>$$

$$\text{s.t.} \quad C \geq \alpha \geq 0 \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

Non-linear problem with feature function:

$$\max_\alpha \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \, \phi(x_i)^T \phi(x_j)$$

$$\text{s.t.} \quad C \geq \alpha \geq 0 \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

- replace x_i with their features $\phi(x)$, aka basis functions

Kernelizing the Linear SVM (2)

Non-linear problem with feature function:

$$\max_\alpha \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \, \phi(x_i)^T \phi(x_j)$$

$$\text{s.t.} \quad C \geq \alpha \geq 0 \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

Non-linear problem with kernel function:

$$\max_\alpha \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \, k(x_i, x_j)$$

$$\text{s.t.} \quad C \geq \alpha \geq 0 \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

- replace inner product $<x_i, x_j>$ with a kernel function $k(x_i, x_j)$ aka covariance function, aka positive definite function

Demo

What is a kernel function?
What is a p.d. kernel function

Note

- So far, we know that kernel function measure similarity.

Feature maps vs kernel functions

Kernel function

- A symmetric (continuous) function $k : \mathcal{X} \times \mathcal{X} \to \Re$, i.e. $k(x, x') = k(x', x)$, is called a *kernel (function)*.

Every feature map ϕ induces a kernel function:

$$k(x, x') = \phi(x)^T \phi(x') = \phi(x')^T \phi(x) = k(x', x)$$

Does every kernel function k induce a feature map?

$$\phi(x) = ?$$

Answer:

- No, only if it fulfills additional requirements (spoiler: positive definiteness)!
- BTW: this topic is part of Functional Analysis.

What is a p.d. kernel function?

Definition 11.3

An $n \times n$ matrix $K \in \mathbb{R}^{n \times n}$ is positive definite, iff for any vector $v \in \mathbb{R}^n$ the inner product of v with its image under K is positive, i.e. $v^T Kv > 0$. If we have $v^T Kv \geq 0$ we call it positive semi-definite.

Definition 11.4

Given data points $x_1, \ldots, x_n \in \mathcal{X}$ and a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, the Gram matrix K is an $n \times n$ matrix with entries defined by the kernel function, i.e. $K_{ij} = k(x_i, x_j)$.

Definition 11.5

A kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive (semi-)definite, iff for any set of data points $x_1, \ldots, x_n \in \mathcal{X}$ the corresponding Gram K is positive (semi-)definite.

Why positive definite?

Notation, notation, changes...

So far:

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{bmatrix}$$

- i.e. each row of X is a data point

In the following example:

$$X = [x_1, x_2 \cdots x_n]$$

- i.e. each column of X is a data point

Why is pos-def a useful property?

Given data $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$.

Linear kernel:

- linear kernel function $k(x, x') = x^T x'$
- calculate kernel matrix $K = X^T X$
- kernel matrix contains all inner products $x_i^T x_j$
- thus $\phi(x) = x$

Nonlinear kernel:

- nonlinear kernel function $k(x, x')$
- calculate kernel matrix with entries $K_{ij} = k(x_i, x_j)$
- for a p.d. kernel function the kernel matrix factorizes $K = Z^T Z$ (see next slide why this is true)
- thus we can read off the feature map e.g. with $Z \in \mathbb{R}^{D \times n}$ and $D >> d$ as $\phi(x_i) = z_i$
- thus, the pd-ness of a kernel function guarantees that the kernel matrix factorizes like this $Z^T Z$

What a minute? Why does K factorize?

- since K is symmetric, it has a Eigenvector-decomposition

$$K = V \Lambda V^T$$

- because k is pd, the kernel matrix K is pd and thus all Eigenvalues are positive, i.e. they can be written as squares,

$$\Lambda_{ii} = \tau_i^2$$

- so we can factorize K as

$$K = (\sqrt{\Lambda} V^T)^T (\sqrt{\Lambda} V^T) = Z^T Z$$

- and read off $phi(x_i) = z_i$ for the data points (with z_i being the the columns of Z

Sketchy summary

Question: What functions are p.d. kernel function?

Short answer (in words)

- kernel function $k(a,b)$ must be symmetric, i.e. $k(a,b) = k(b,a)$
- kernel function must be positive definite, i.e. ...
- kernel functions are for functions, what pos. def. matrices are for matrices
- in Functional Analysis we study functions instead of vectors, linear operators instead of matrices
- kernel functions are pos. def. linear operators
- (analogous to: certain matrices are pos. def. linear mappings)

End of the SVM Section

Appendix

Linear algebra reminder: SVD and EVD

Theorem 11.6 (Singular value decomposition/SVD)

Any matrix X can be factorized into three matrices, i.e. can be written as a product of three matrices

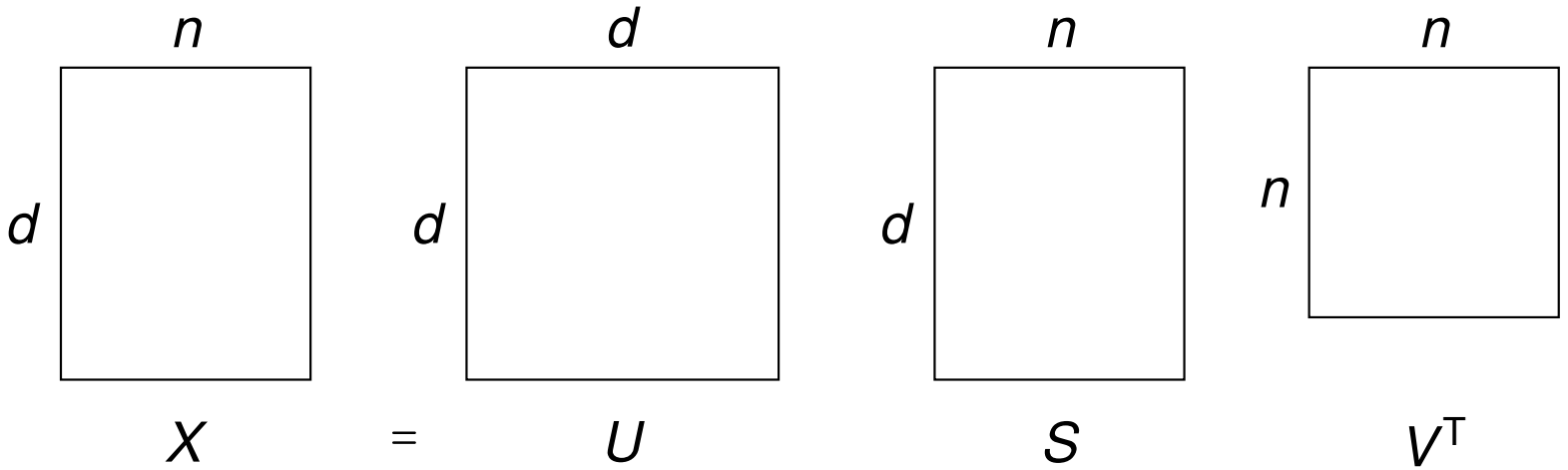
$$X = USV^T$$

such that

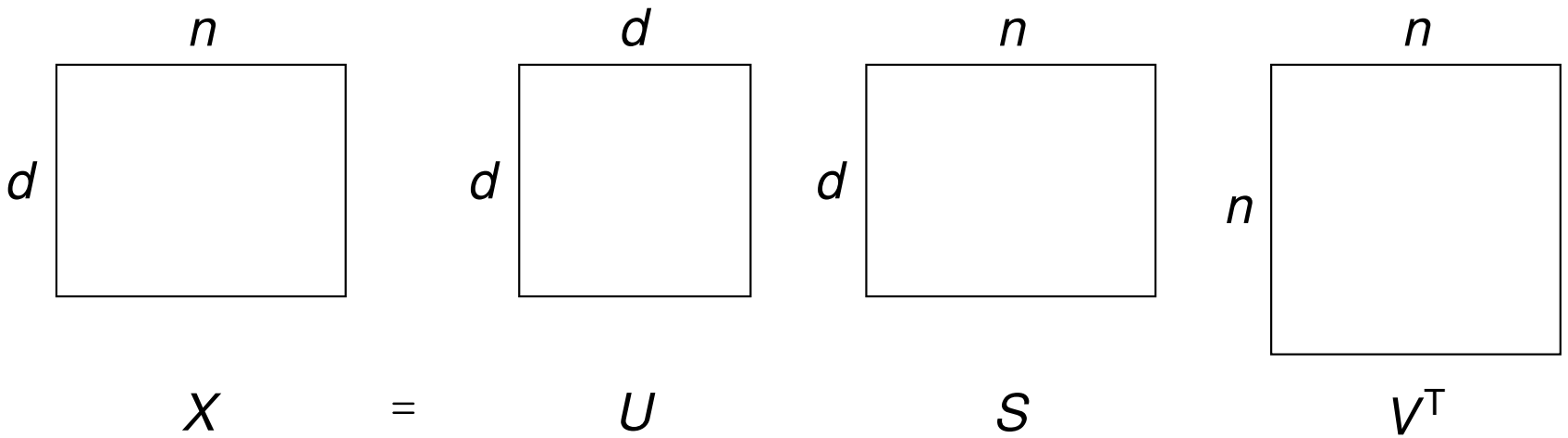
- *U is $d \times d$, and V is $n \times n$, so S is rectangular of size $d \times n$.*
- *U and V are unitary, i.e. $UU^T = I_d$ and $VV^T = I_n$*
- *I_d and I_n being $n \times n$ and $d \times d$ dimensional identity matrices*
- *S is diagonal matrix, entries are called singular values (SVs)*

SVD — graphically

Case (i): $d \geq n$

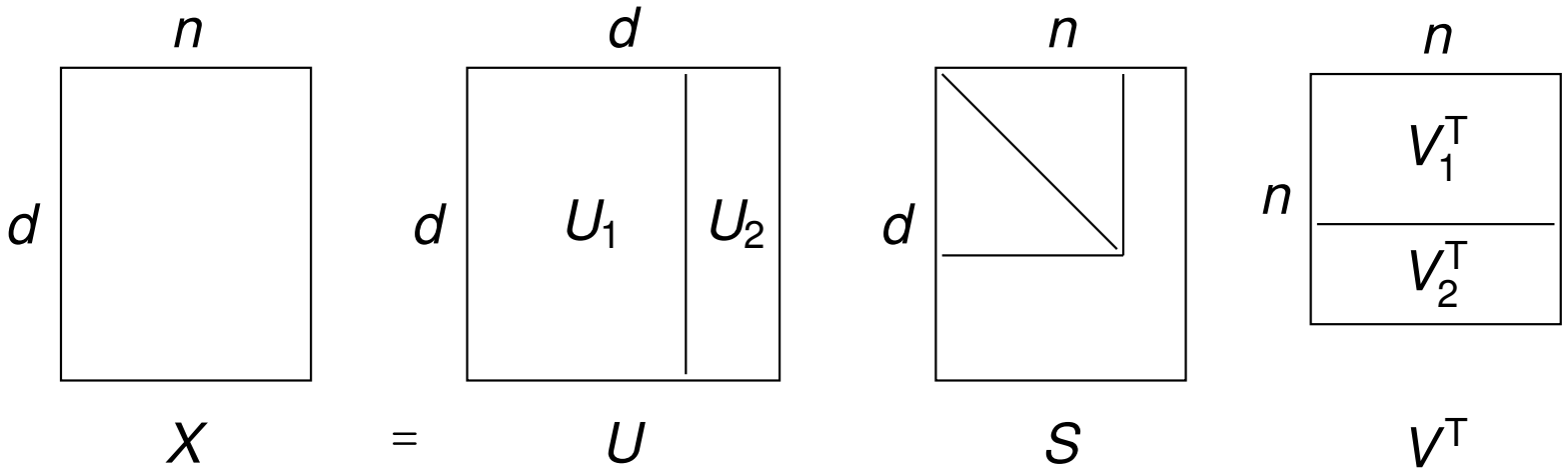


Case (ii): $d < n$

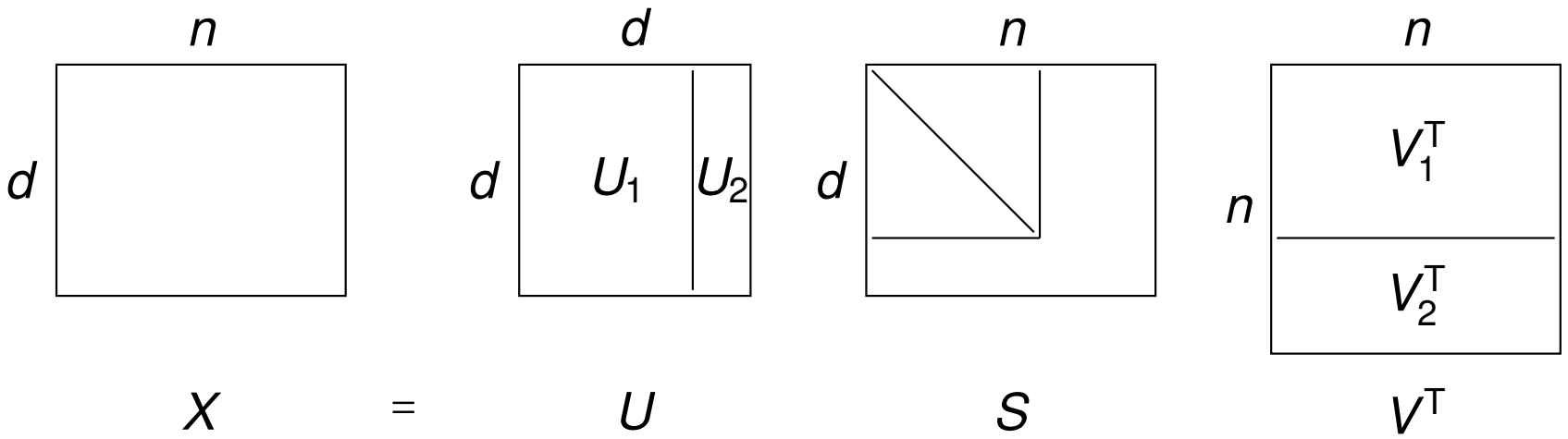


SVD — range and null space

Case (i): $d \geq n$



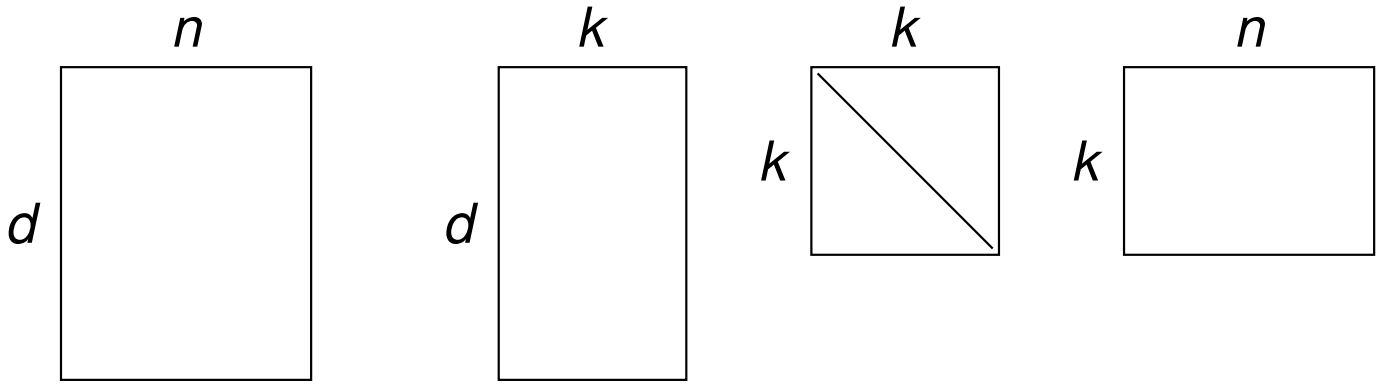
Case (ii): $d < n$



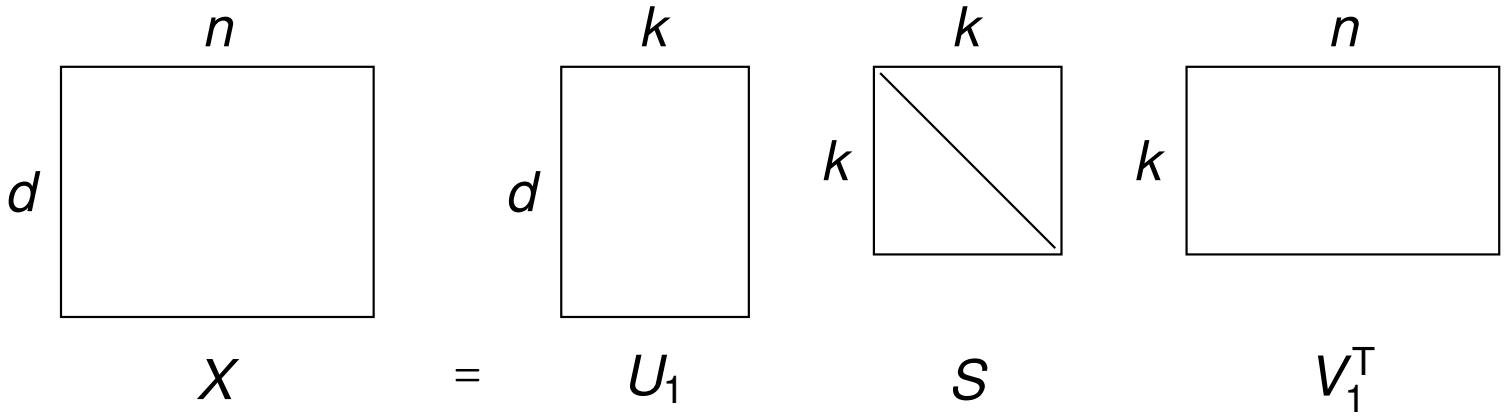
Null space of X is V_2. *Range* of X is U_1.

SVD — economy size and rank

Case (i): $d \geq n$



Case (ii): $d < n$



rank of X is k, the number of non-zero singular values.

Theorem 11.7 (Eigenvalue decomposition/EVD)

Any symmetric matrix A can be decomposed into

$$A = V \Lambda V^T$$

- *with Λ being diagonal matrix*
- *with V being unitary matrix, i.e. $VV^T = V^T V = I$*
- *values along the diagonal of Λ are called eigenvalues*
- *columns of V are called eigenvectors*

Note

- for eigenvalue λ and eigenvector v, the factorization $A = V \Lambda V^T$ implies $Av = \lambda v$
- EVs stands for “eigenvectors” or for “eigenvalues” depending on the contexts...

Are EVD and SVD related?

YES!

(Economy-sized) SVD of a (rectangular) data matrix X:

$$X = USV^T.$$

Calculate XX^T and X^TX:

$$XX^T = USV^T VSU^T = US^2U^T = U\Lambda U^T$$
$$X^TX = VSU^T USV^T = VS^2V^T = V\Lambda V^T$$

- left singular vectors U of X are the eigenvectors of XX^T
- right singular vectors V of X are the eigenvectors of X^TX
- the squared SVs of X are the eigenvalues of XX^T *and* of X^TX

Can we “understand” the feature space? What do we know about it?

Representations of the feature space for given data (1)

Theorem 11.8 (Eigenvector decomposition of PD matrix)

If an $n \times n$ (symmetric) matrix K is positive definite, then there exist eigenvectors $v_1, \ldots, v_n \in \mathbb{R}^n$ and positive eigenvalues $\lambda_1, \ldots, \lambda_n \geq 0$ such that K can be rewritten as

$$K = \sum_{i=1}^{n} \lambda_i v_i v_i^T = V \Lambda V^T,$$

where $v_i v_i^T \in \mathbb{R}^{n \times n}$ is the outer product and $V = [v_1, \ldots, v_n]$ is the matrix with the eigenvectors as columns and Λ the diagonal matrix with the eigenvalues along its diagonal.

- given (training) data $x_1, \ldots, x_n \in \mathcal{X}$
- k PD implies the gram matrix K is PD
- thus $K = V \Lambda V^T = Z^T Z$ with $Z = \Lambda^{1/2} V^T$
- define feature map $\phi(x_i) := Z_{:i}$ as the ith column of Z
- however, this construction can not map arbitrary (test) points x

Note on positive definiteness

Q: are pos def matrices symmetric?

- Typically, pos def is only defined for symmetric (Hermitian) matrices, i.e. $A = A^H = \overline{A^T}$.
- Several possible reasons:
  1. if a matrix is not Hermitian, its eigenvalues can be complex. then what is positivity?
  2. for a non-symmetric matrix A, consider quadratic form $f_A(x) = x^T Ax$. Note that $f_A = f_{(A+A^T)/2}$. Thus pos def of f_A is same as pos def of symmetric $(A + A^T)/2$.
  3. for complex pos def matrix A, A Hermitian is equiv. to $x^T Ax$ being real....

Sources:

- `https://math.stackexchange.com/questions/1107230/why-is-positive-semi-definite-only-defined-for-symmetric-matrices`
- `https://math.stackexchange.com/questions/516533/symmetric-vs-positive-semidefinite?rq=1`

Representations of the feature space for given data (2)

Given data $x_1, \ldots, x_n \in \mathcal{X}$.

Definition 11.9 (Empirical kernel map)

$$\begin{array}{rccc} \phi_n : & \mathcal{X} & \longrightarrow & \mathbb{R}^n \\ & x & \longmapsto & \begin{bmatrix} k(x_1, x) \\ \vdots \\ k(x_n, x) \end{bmatrix} \end{array}$$

Unfortunately:

$$k(x, x') \neq \phi_n(x)^\mathsf{T} \phi_n(x').$$

But we have:

$$k(x, x') = \phi_n(x)^\mathsf{T} K^{-1} \phi_n(x').$$

Representations of the feature space for given data (3)

Given data $x_1, \ldots, x_n \in \mathcal{X}$.

Definition 11.10 (Kernel PCA map)

$$\begin{array}{rccc} \phi: & \mathcal{X} & \longrightarrow & \mathbb{R}^n \\ & x & \longmapsto & K^{-1/2} \begin{bmatrix} k(x_1, x) \\ \vdots \\ k(x_n, x) \end{bmatrix} \end{array}$$

- ϕ is a feature map for k

$$k(x, x') = \phi_n(x)^{\mathsf{T}} K^{-1} \phi_n(x').$$

Representations of the feature space without data (1)

Notation:

$$3 := \{0, 1, 2\}$$
$$d := \{0, 1, \ldots, d-1\}$$

Vectors as a mappings:

$$\begin{array}{rccc} v: & d & \longrightarrow & \mathbb{R} \\ & i & \longmapsto & v_i \end{array}$$

Sets of functions:

$$\mathbb{R}^d = \{f : d \to \mathbb{R}\}$$
$$\mathbb{R}^{\mathcal{X}} = \{f : \mathcal{X} \to \mathbb{R}\}$$

Kernel functions as continuous generalizations of matrices:

$$\mathbb{R}^{n \times n} = \{f : n \times n \to \mathbb{R}\}$$
$$\mathbb{R}^{\mathcal{X} \times \mathcal{X}} = \{f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}\}$$

Representations of the feature space without data (2)

Generalizing eigenvector decomposition to functions:

Theorem 11.11 (Mercer)

If a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive definite, then there exist eigenfunctions $\phi_1, \phi_2, \ldots : \mathcal{X} \to \mathbb{R}$ and positive eigenvalues $\lambda_1, \lambda_2, \ldots \geq 0$ such that k can be rewritten as:

$$k(x, x') = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(x').$$

Definition 11.12 (Mercer kernel map)

$$\begin{array}{rccc} \phi: & \mathcal{X} & \longrightarrow & \mathbb{R}^\infty \\ & x & \longmapsto & \begin{bmatrix} \sqrt{\lambda_1}\,\phi_1(x) \\ \sqrt{\lambda_2}\,\phi_2(x) \\ \vdots \end{bmatrix}. \end{array}$$

Note that $\mathbb{R}^\infty = \mathbb{R}^{\mathbb{N}}$ and $\langle \phi(x), \phi(x') \rangle = k(x, x')$ by Mercer's theorem.

Note

- Mercer's theorem doesn't give us a recipe to calculate ϕ
- AFAIK (as far as I know) it is only a result about the existence of ϕ
- if you want to be sure about this, check the proof, whether it is constructive, i.e. whether it provides the general recipe...

Representations of the feature space without data (3)

- Mercer kernel map maps to countably infinite dimensional space
- next we define a feature space of functions

Definition 11.13 (Reproducing kernel map)

$$\begin{array}{rccc} \phi: & \mathcal{X} & \longrightarrow & \mathbb{R}^{\mathcal{X}} \\ & x' & \longmapsto & k(\cdot, x') \end{array}$$

where $k(\cdot, x')$ is k curried, i.e.

$$\begin{array}{rccc} k(\cdot, x'): & \mathcal{X} & \longrightarrow & \mathbb{R} \\ & x & \longmapsto & k(x, x'). \end{array}$$

- $\mathbb{R}^{\mathcal{X}}$ is the space of functions from $\mathcal{X}$ to $\mathbb{R}$
- $\phi(\mathcal{X})$ is the feature space image of the input space

Representations of the feature space without data (4)

Question:

- How can we turn $\phi(\mathcal{X}) \subset \mathbb{R}^{\mathcal{X}}$ into a Hilbert space?

Three steps:

1. Turn $\phi(\mathcal{X})$ into a vector space $\mathcal{H}$.
2. Define dot-product in $\mathcal{H}$.
3. Complete $\mathcal{H}$ to a Hilbert space.

Ad 1.

Given data points $x_1, \ldots, x_m \in \mathcal{X}$ and coefficients $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$ let

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

Thus for $x' \in \mathcal{X}$ we have $\phi(x') = k(\cdot, x') \in \mathcal{H}$.

Representations of the feature space without data (5)

- consider two function $f, g \in \mathcal{H}$:

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) \qquad g(\cdot) = \sum_{j=1}^{m'} \beta_i k(\cdot, x'_i)$$

- linear combinations: $\gamma f(\cdot) + g(\cdot) \in \mathcal{H}$
- define dot product in $\mathcal{H}$:

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

Theorem 11.14

$\langle f, g \rangle$ is a dot product, i.e.

(i) $\langle f, g \rangle = \langle g, f \rangle$ — *symmetry*

(ii) $\langle af_1 + bf_2, g \rangle = a\langle f_1, g \rangle + b\langle f_2, g \rangle$ *and* $\langle f, ag_1 + bg_2 \rangle = a\langle f, g_1 \rangle + b\langle f, g_2 \rangle$ — *bilinearity*

(iii) $\langle f, f \rangle = 0$ *implies* $f(x) = 0$ *for any* $x \in \mathcal{X}$ — *strictly positive definite*

Representations of the feature space without data (6)

To show (iii) we need a lemma:

Lemma 11.15

(iv) $\langle k(\cdot, x), g\rangle = g(x)$ — *k is representer of evaluation*

(v) $\langle k(\cdot, x), k(\cdot, x')\rangle = k(x, x')$ — *k is reproducing*

(vi) $\langle f, f\rangle \geq 0$

(vii) $\langle \cdot, \cdot \rangle$ *is positive definite kernel function for two elements of the function space* $\mathbb{R}^{\mathcal{X}}$.

(viii) $|k(x, x')|^2 \leq k(x, x)k(x', x')$ — *Cauchy-Schwartz for kernels*

(ix) $|\langle f, g\rangle|^2 \leq \langle f, f\rangle\langle g, g\rangle$

Proof: To show (iv) note that the dot product can be rewritten with the definitions of f and g,

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

Replacing in the left equation f by $k(\cdot, x)$ we showed (iv). Replacing g in (iv) by $k(\cdot, x')$ we showed (v).
(vi) follows from the PDness of k, since it implies the following inequality:

$$\langle f, f \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) = \alpha^T K \alpha \geq 0$$

where $\alpha = [\alpha_1, \ldots, \alpha_m]^T$ is the vector of coefficients of f and the data points $x_1, \ldots, x_m \in \mathcal{X}$ induce the Gram matrix K.

Proof continued:

(vii) appears at first glance a bit weird. It is only of technical importance to show that the Cauchy-Schwartz inequality holds for the dot-product. To show (vii) we follow the definition of PDness of kernel functions. We consider functions $f_1, \ldots, f_n \in \mathrm{H}$ and define the $n \times n$ Gram matrix G with entries $G_{ij} = \langle f_i, f_j \rangle$. For any vector $v \in \mathbb{R}^n$ we have to show $v^\mathsf{T} G v \geq 0$,

$$v^\mathsf{T} G v = \sum_{i=1}^{n} \sum_{j=1}^{n} v_i v_j G_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{n} v_i v_j \langle f_i, f_j \rangle = \langle \sum_{i=1}^{n} v_i f_i, \sum_{j=1}^{n} v_j f_j \rangle \geq 0$$

where we used the bilinearity of the dot product, and (vi) for the inequality. We have shown that (vii) for the only reason to prove (ix) which follows from (viii) and (vii). For brevity we do not show a proof of (viii).

Proof (continued):
Having fully proven the lemma, we can now prove (iii). For arbitrary $x \in \mathcal{X}$ and $f \in \mathbb{R}^{\mathcal{X}}$,

(...) exercise

Thus $f(x) = 0$, which completes the proof.

Back to $\mathcal{H}$:
Up to now $\mathcal{H}$ is a pre-Hilbert space (a vector space with dot-product). To complete $\mathcal{H}$ to a Hilbert space, limits must be added to it and the dot-product appropriately extended. ...

End of appendix