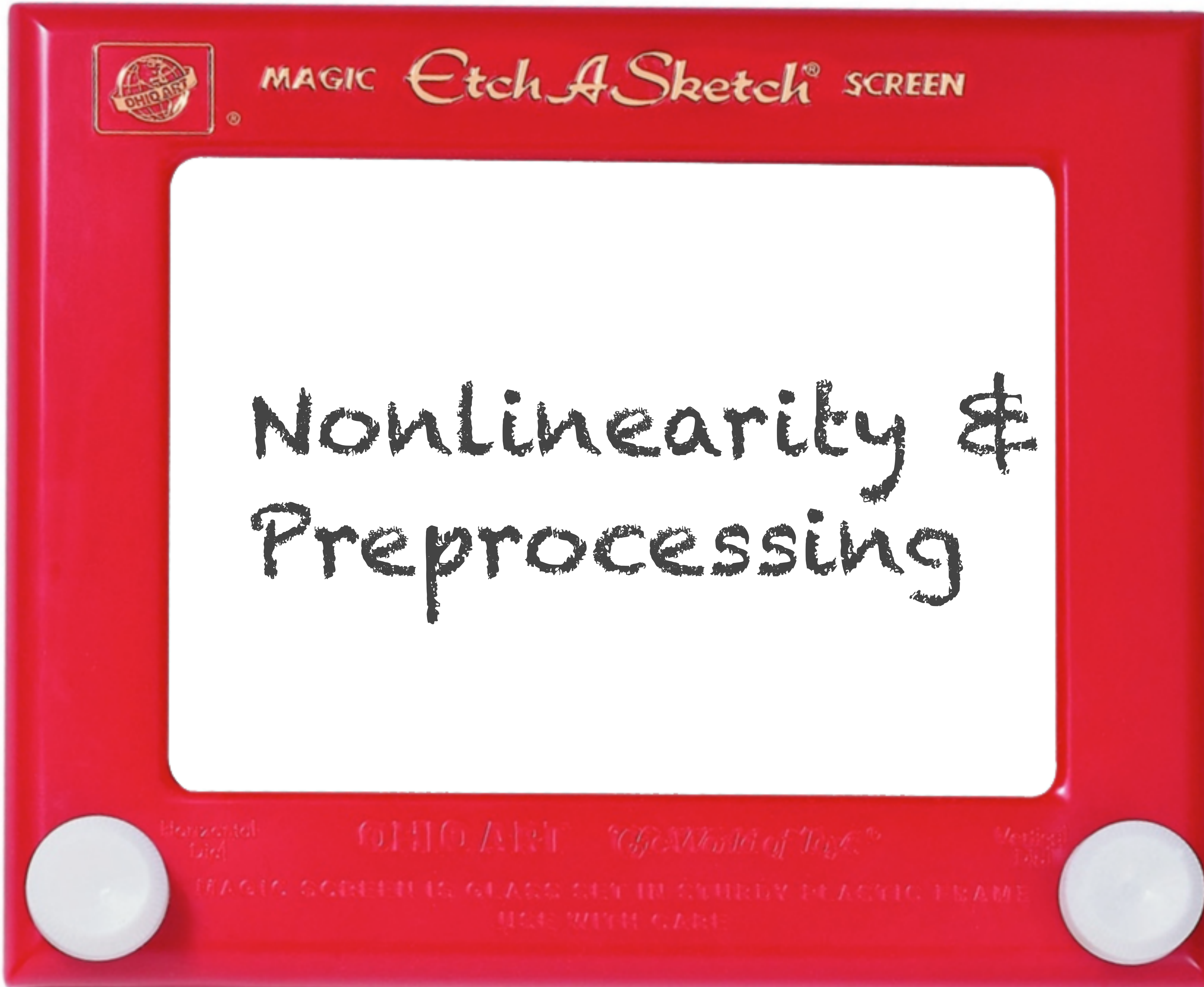
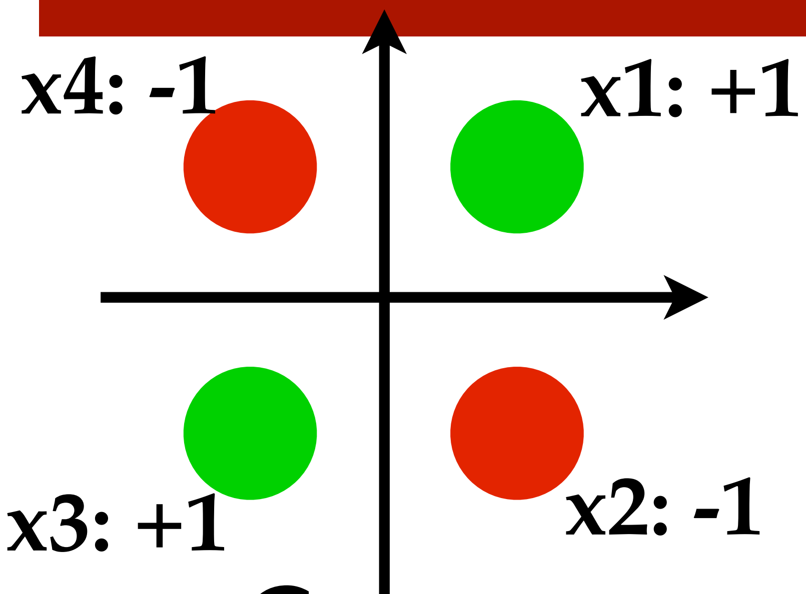
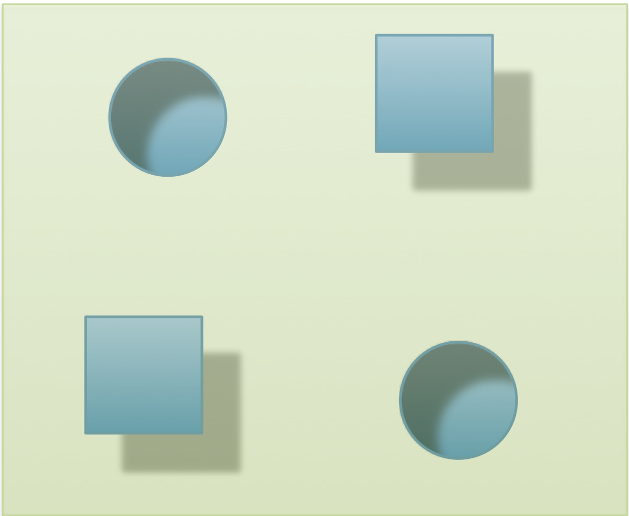


Nonlinearity & Preprocessing

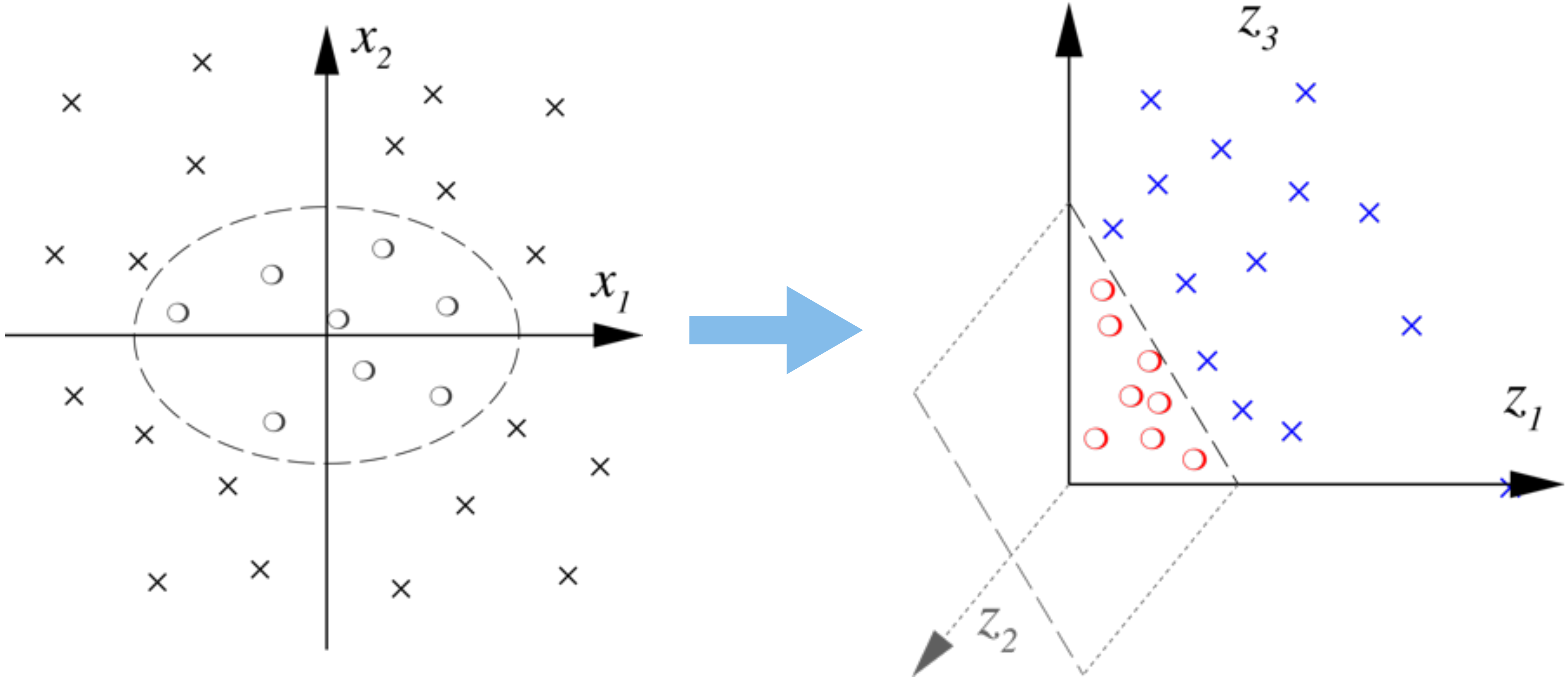
Nonlinear Features





- Concatenated (combined) features
  - XOR: $x = (x_1, x_2, x_1x_2)$
  - income: add "degree + major"
- Perceptron
  - Map data into feature space $x \to \phi(x)$
  - Solution in span of $\phi(x_i)$

Quadratic Features



- Separating surfaces are Circles, hyperbolae, parabolae

Constructing Features (very naive OCR system)

	1	2	3	4	5	6	7	8	9	0
Loops	0	0	0	1	0	1	0	2	1	1
3 Joints	0	0	0	0	0	1	0	0	1	0
4 Joints	0	0	0	1	0	0	0	1	0	0
Angles	0	1	1	1	1	0	1	0	0	0
Ink	1	2	2	2	2	2	1	3	2	2

Feature Engineering for Spam Filtering

- bag of words
- pairs of words
- date & time
- recipient path
- IP number
- sender
- encoding
- links
- combinations of above

```
Delivered-To: alex.smola@gmail.com
Received: by 10.216.47.73 with SMTP id s51cs361171web;
        Tue, 3 Jan 2012 14:17:53 -0800 (PST)
Received: by 10.213.17.145 with SMTP id s17mr2519891eba.147.1325629071725;
        Tue, 03 Jan 2012 14:17:51 -0800 (PST)
Return-Path: <alex+caf_=alex.smola=gmail.com@smola.org>
Received: from mail-ey0-f175.google.com (mail-ey0-f175.google.com [209.85.215.175])
        by mx.google.com with ESMTPS id n4si29264232eef.57.2012.01.03.14.17.51
        (version=TLSv1/SSLv3 cipher=OTHER);
        Tue, 03 Jan 2012 14:17:51 -0800 (PST)
Received-SPF: neutral (google.com: 209.85.215.175 is neither permitted nor denied by best
guess record for domain of alex+caf_=alex.smola=gmail.com@smola.org) client-
ip=209.85.215.175;
Authentication-Results: mx.google.com; spf=neutral (google.com: 209.85.215.175 is neither
permitted nor denied by best guess record for domain of alex
+caf_=alex.smola=gmail.com@smola.org) smtp.mail=alex+caf_=alex.smola=gmail.com@smola.org;
dkim=pass (test mode) header.i=@googlemail.com
Received: by eaal1 with SMTP id l1so15092746eaa.6
        for <alex.smola@gmail.com>; Tue, 03 Jan 2012 14:17:51 -0800 (PST)
Received: by 10.205.135.18 with SMTP id ie18mr5325064bkc.72.1325629071362;
        Tue, 03 Jan 2012 14:17:51 -0800 (PST)
X-Forwarded-To: alex.smola@gmail.com
X-Forwarded-For: alex@smola.org alex.smola@gmail.com
Delivered-To: alex@smola.org
Received: by 10.204.65.198 with SMTP id k6cs206093bki;
        Tue, 3 Jan 2012 14:17:50 -0800 (PST)
Received: by 10.52.88.179 with SMTP id bh19mr10729402vdb.38.1325629068795;
        Tue, 03 Jan 2012 14:17:48 -0800 (PST)
Return-Path: <althoff.tim@googlemail.com>
Received: from mail-vx0-f179.google.com (mail-vx0-f179.google.com [209.85.220.179])
        by mx.google.com with ESMTPS id dt4si11767074vdb.93.2012.01.03.14.17.48
        (version=TLSv1/SSLv3 cipher=OTHER);
        Tue, 03 Jan 2012 14:17:48 -0800 (PST)
Received-SPF: pass (google.com: domain of althoff.tim@googlemail.com designates
209.85.220.179 as permitted sender) client-ip=209.85.220.179;
Received: by vcbf13 with SMTP id f13so11295098vcb.10
        for <alex@smola.org>; Tue, 03 Jan 2012 14:17:48 -0800 (PST)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
        d=googlemail.com; s=gamma;
        h=mime-version:sender:date:x-google-sender-auth:message-id:subject
         :from:to:content-type;
        bh=WCbdZ5sXac25dpH02XcRyDOdts993hKwsAVXpGrFh0w=;
        b=WK2B2+ExWnf/gvTkw6uUvKuP4XeoKnlJq3USYTm0RARK8dSFjyOQsIHeAP9Yssxp6O
         7ngGoTzYqd+ZsyJfvQcLAWp1PCJhG8AMcnqWkx0NMeoFvIp2HQooZwxSOCx5ZRgY+7qX
         uIbbdna4lUDXj6UFe16SpLDCkptd8OZ3gr7+o=
MIME-Version: 1.0
Received: by 10.220.108.81 with SMTP id e17mr24104004vcp.67.1325629067787;
 Tue, 03 Jan 2012 14:17:47 -0800 (PST)
Sender: althoff.tim@googlemail.com
Received: by 10.220.17.129 with HTTP; Tue, 3 Jan 2012 14:17:47 -0800 (PST)
Date: Tue, 3 Jan 2012 14:17:47 -0800
X-Google-Sender-Auth: 6bwi6D17HjZIkxOEol38NZzyeHs
Message-ID: <CAFJJHDGPBW+SdZg0MdAABiAKydDk9tpeMoDijYGjoGO-WC7osg@mail.gmail.com>
Subject: CS 281B. Advanced Topics in Learning and Decision Making
From: Tim Althoff <althoff@eecs.berkeley.edu>
To: alex@smola.org
Content-Type: multipart/alternative; boundary=f46d043c7af4b07e8d04b5a7113a

--f46d043c7af4b07e8d04b5a7113a
Content-Type: text/plain; charset=ISO-8859-1
```

The Perceptron on features

initialize $w = 0$
repeat
  Pick (x_i, y_i) from data
  if $y_i(w \cdot \Phi(x_i)) \leq 0$ then

$$w' = w + y_i \Phi(x_i)$$

until $y_i(w \cdot \Phi(x_i)) > 0$ for all i

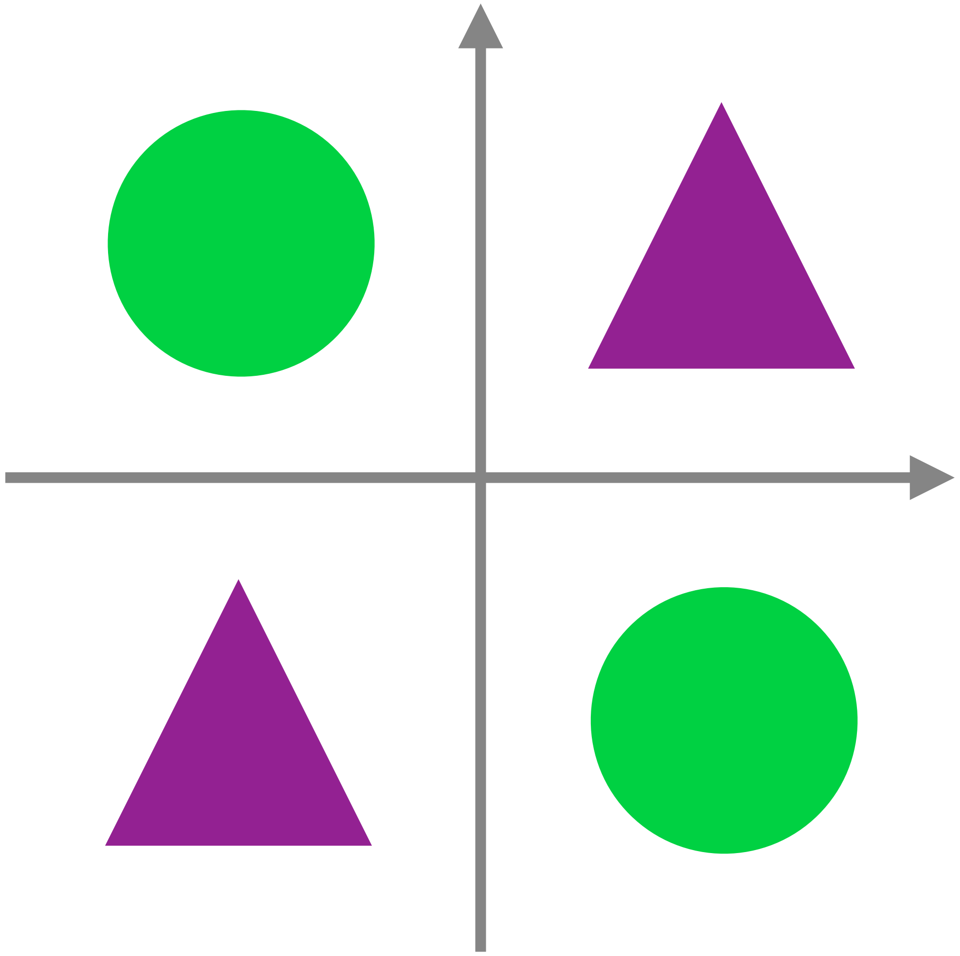
- Nothing happens if classified correctly
- Weight vector is linear combination $w = \sum_{i \in I} \alpha_i \phi(x_i)$
- Classifier is (implicitly) a linear combination of inner products $f(x) = \sum_{i \in I} \alpha_i \langle \phi(x_i), \phi(x) \rangle$

Problems

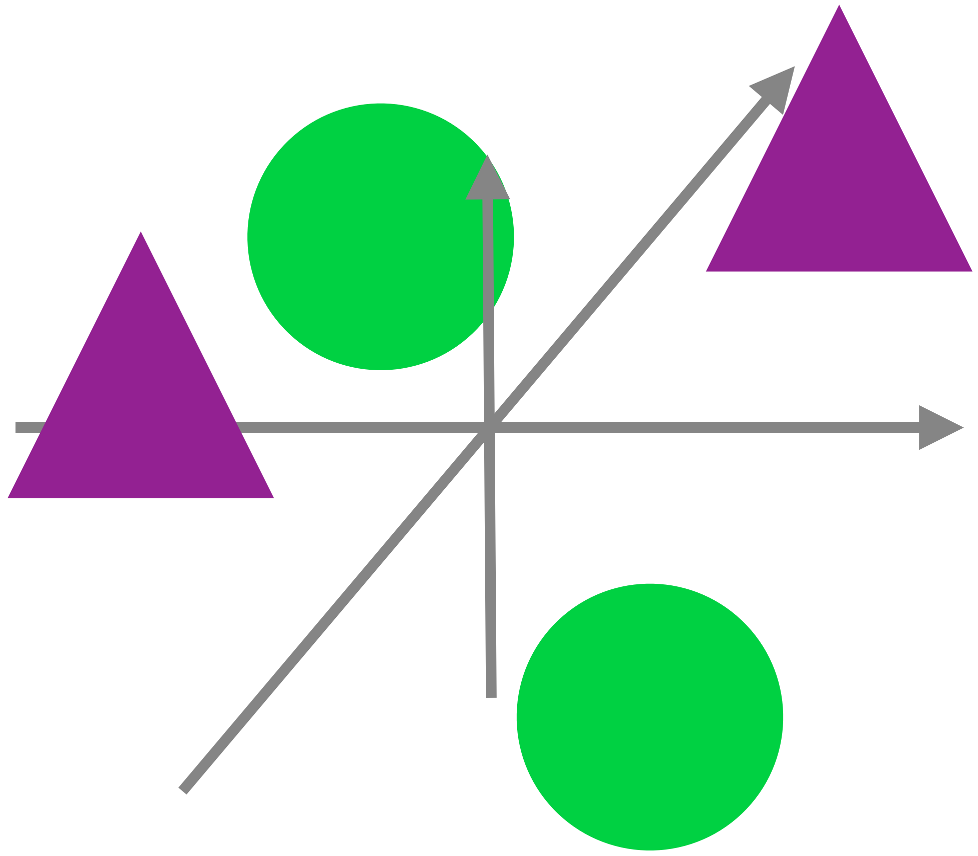
- Problems
  - Need domain expert (e.g. Chinese OCR)
  - Often expensive to compute
  - Difficult to transfer engineering knowledge
- Shotgun Solution
  - Compute many features
  - Hope that this contains good ones
  - How to do this efficiently?

Kernels

Solving XOR



(x_1, x_2)



$(x_1, x_2, x_1 x_2)$

- XOR not linearly separable
- Mapping into 3 dimensions makes it easily solvable

SVM with a polynomial Kernel visualization

Created by:
Udi Aharoni

Kernels as dot products

Problem

- Extracting features can sometimes be very costly.
- Example: second order features in 1000 dimensions. This leads to $5 \cdot 10^5$ numbers. For higher order polynomial features much worse.

Solution

Don't compute the features, try to compute dot products implicitly. For some features this works . . .

Definition

A kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a symmetric function in its arguments for which the following property holds

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle \text{ for some feature map } \Phi.$$

If $k(x, x')$ is much cheaper to compute than $\Phi(x)$ . . .

Quadratic Kernel

Quadratic Features in $\mathbb{R}^2$

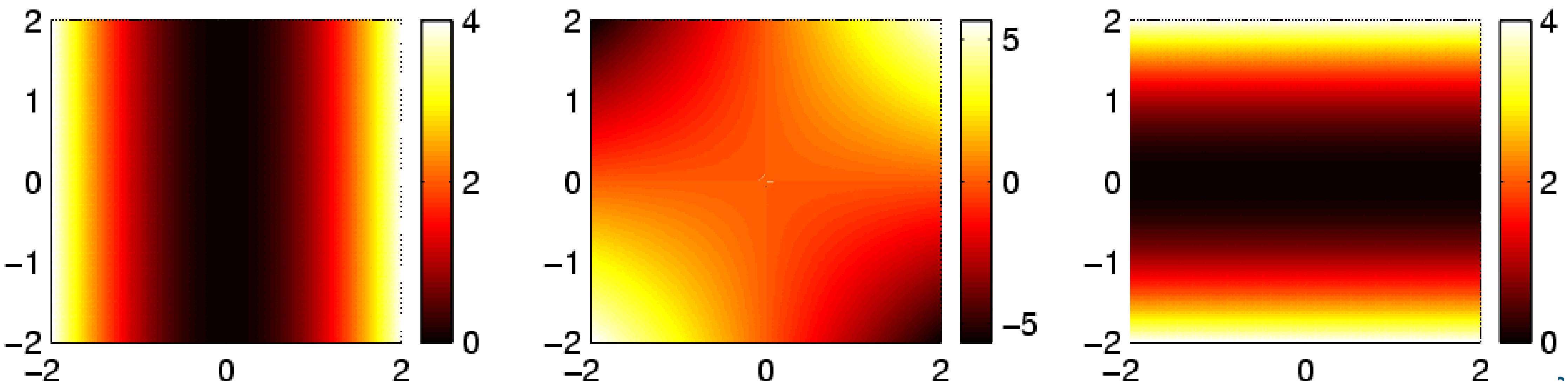
$$\Phi(x) := \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$$

Dot Product

$$\begin{aligned}\langle \Phi(x), \Phi(x') \rangle &= \left\langle \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right), \left({x'_1}^2, \sqrt{2}x'_1x'_2, {x'_2}^2\right) \right\rangle \\ &= \langle x, x' \rangle^2.\end{aligned}$$

Insight

Trick works for any polynomials of order d via $\langle x, x' \rangle^d$.



Kernelized Perceptron

initialize $f = 0$ **Functional Form**

repeat

  Pick (x_i, y_i) from data

  if $y_i f(x_i) \leq 0$ then

    $f(\cdot) \leftarrow f(\cdot) + y_i k(x_i, \cdot)$

until $y_i f(x_i) > 0$ for all i

- Nothing happens if classified correctly
- Weight vector is linear combination $w = \sum_{i \in I} \alpha_i \phi(x_i)$
- Classifier is linear combination of inner products

$$f(x) = \sum_{i \in I} \alpha_i \langle \phi(x_i), \phi(x) \rangle = \sum_{i \in I} \alpha_i k(x_i, x)$$

Kernelized Perceptron

Primal Form

update weights

$$w \leftarrow w + y_i \phi(x_i)$$

classify

$$f(k) = w \cdot \phi(x)$$

Dual Form

update linear coefficients

$$\alpha_i \leftarrow \alpha_i + y_i$$

implicitly equivalent to:

$$w = \sum_{i \in I} \alpha_i \phi(x_i)$$

- Nothing happens if classified correctly
- Weight vector is linear combination $\boxed{w = \sum_{i \in I} \alpha_i \phi(x_i)}$
- Classifier is linear combination of inner products

$$f(x) = \sum_{i \in I} \alpha_i \langle \phi(x_i), \phi(x) \rangle = \sum_{i \in I} \alpha_i k(x_i, x)$$

Kernelized Perceptron

Primal Form

update weights

$$w \leftarrow w + y_i \phi(x_i)$$

classify

$$f(k) = w \cdot \phi(x)$$

Dual Form

update linear coefficients

$$\alpha_i \leftarrow \alpha_i + y_i$$

implicitly equivalent to:

$$w = \sum_{i \in I} \alpha_i \phi(x_i)$$

classify

$$f(x) = w \cdot \phi(x) = [\sum_{i \in I} \alpha_i \phi(x_i)] \phi(x)$$

hard!

$$= \sum_{i \in I} \alpha_i \langle \phi(x_i), \phi(x) \rangle$$

easy!

$$= \sum_{i \in I} \alpha_i k(x_i, x)$$

Kernelized Perceptron

initialize $\alpha_i = 0$ for all i

repeat

  Pick (x_i, y_i) from data

    if $y_i f(x_i) \leq 0$ then

      $\alpha_i \leftarrow \alpha_i + y_i$

until $y_i f(x_i) > 0$ for all i

if #features > #examples,
dual is easier;
otherwise primal is easier

Dual Form

update linear coefficients

$\alpha_i \leftarrow \alpha_i + y_i$

implicitly

$$w = \sum_{i \in I} \alpha_i \phi(x_i)$$

classify

$$f(x) = w \cdot \phi(x) = [\sum_{i \in I} \alpha_i \phi(x_i)] \phi(x)$$

hard!

$$= \sum_{i \in I} \alpha_i \langle \phi(x_i), \phi(x) \rangle$$

easy!

$$= \sum_{i \in I} \alpha_i k(x_i, x)$$

Kernelized Perceptron

Primal Perceptron

update weights

$$w \leftarrow w + y_i \phi(x_i)$$

classify

$$f(k) = w \cdot \phi(x)$$

Dual Perceptron

update linear coefficients

$$\alpha_i \leftarrow \alpha_i + y_i$$

implicitly

$$w = \sum_{i \in I} \alpha_i \phi(x_i)$$

if #features >> #examples,
dual is easier;
otherwise primal is easier

Q: when is #features >> #examples?

A: higher-order polynomial kernels
or exponential kernels (inf. dim.)

Kernelized Perceptron

Pros/Cons of Kernel in Dual

- pros:
  - no need to store long feature and weight vectors (memory)
- cons:
  - sum over all misclassified training examples for test (time)
  - need to store all misclassified training examples (memory)
    - called "support vector set"
    - SVM will minimize this set!

Dual Perceptron

update linear coefficients

$$\alpha_i \leftarrow \alpha_i + y_i$$

implicitly

$$w = \sum_{i \in I} \alpha_i \phi(x_i)$$

classify

$$f(x) = w \cdot \phi(x) = [\sum_{i \in I} \alpha_i \phi(x_i)] \phi(x)$$

$$= \sum_{i \in I} \alpha_i \langle \phi(x_i), \phi(x) \rangle$$

hard!

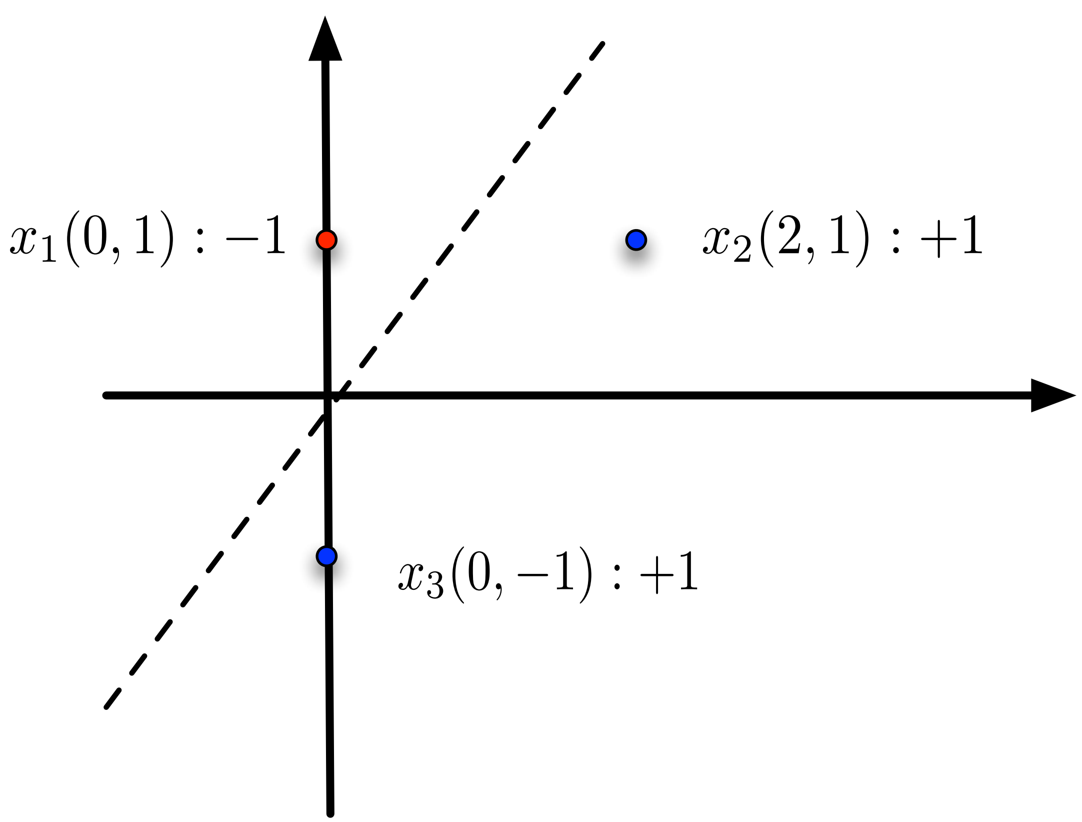
$$= \sum_{i \in I} \alpha_i k(x_i, x)$$

easy!

Kernelized Perceptron

Primal Perceptron

update on	new param.
x1: -1	w = (0, -1)
x2: +1	w = (2, 0)
x3: +1	w = (2, -1)



Dual Perceptron

update on	new param.	w (implicit)
x1: -1	α = (-1, 0, 0)	-x1
x2: +1	α = (-1, 1, 0)	-x1 + x2
x3: +1	α = (-1, 1, 1)	-x1 + x2 + x3

linear kernel (identity map)

final implicit w = (2, -1)

geometric interpretation

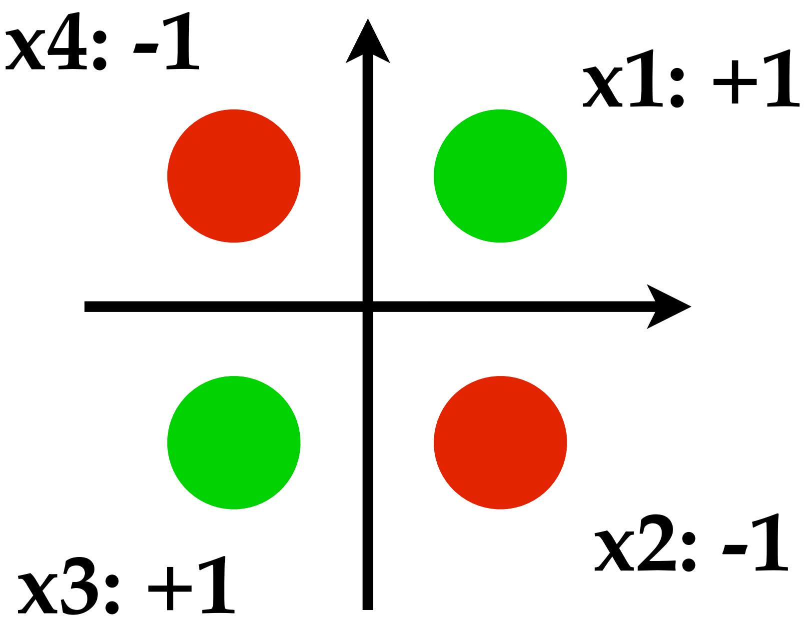
of dual classification:

sum of dot-products with x2 & x3

bigger than dot-product with x1

(agreement w/ positive > w/ negative)

XOR Example



Dual Perceptron

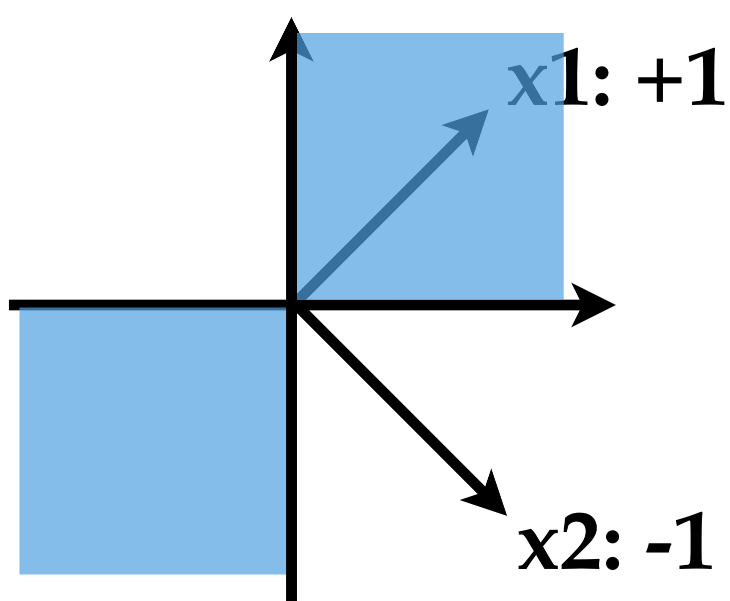
update on	new param.	w (implicit)
x1: +1	$\alpha = (+1, 0, 0, 0)$	$\phi(x1)$
x2: -1	$\alpha = (+1, -1, 0, 0)$	$\phi(x1) - \phi(x2)$

$$k(x, x') = (x \cdot x')^2 \quad \Leftrightarrow \quad \phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

$$w = (0, 0, 2\sqrt{2})$$

classification rule in dual/geom:

$$(x \cdot \mathbf{x}_1)^2 > (x \cdot \mathbf{x}_2)^2$$
$$\Rightarrow \cos^2\theta_1 > \cos^2\theta_2$$
$$\Rightarrow |\cos\theta_1| > |\cos\theta_2|$$

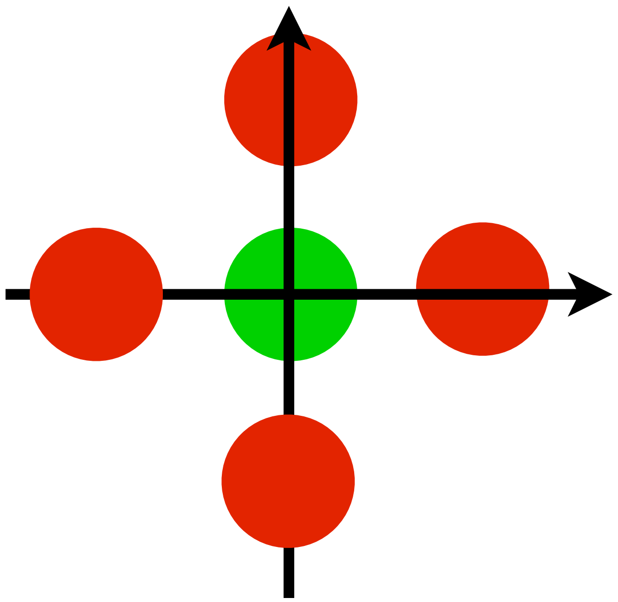


in dual/algebra:

$$(x \cdot \mathbf{x}_1)^2 > (x \cdot \mathbf{x}_2)^2$$
$$\Rightarrow (x_1 + x_2)^2 > (x_1 - x_2)^2$$
$$\Rightarrow x_1x_2 > 0$$

also verify in primal

Circle Example??



Dual Perceptron

update on	new param.	**w** (implicit)
x1: +1	**α = (+1, 0, 0, 0)**	**ϕ(x1)**
x2: -1	**α = (+1, -1, 0, 0)**	**ϕ(x1) - ϕ(x2)**

$$k(x, x') = (x \cdot x')^2 \;\Leftrightarrow\; \phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

Polynomial Kernels

Idea

- We want to extend $k(x, x') = \langle x, x' \rangle^2$ to

$$k(x, x') = (\langle x, x' \rangle + c)^d \text{ where } c > 0 \text{ and } d \in \mathbb{N}.$$

- Prove that such a kernel corresponds to a dot product.

Proof strategy

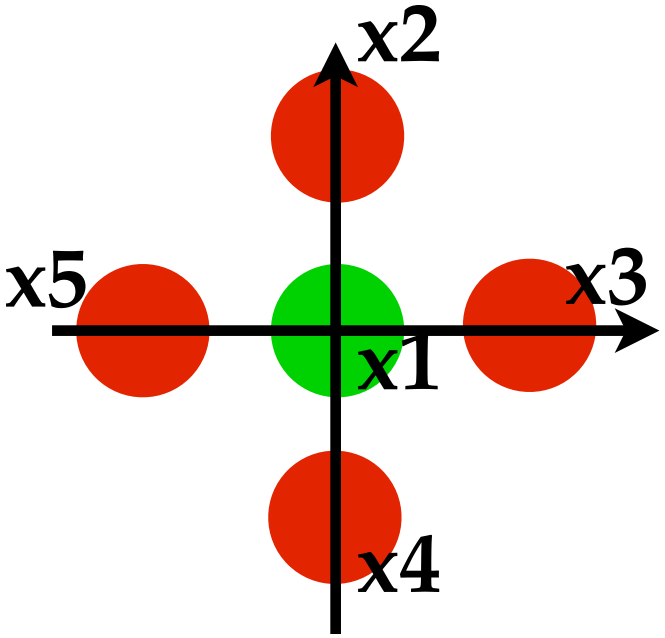
Simple and straightforward: compute the explicit sum given by the kernel, i.e.

+c is just augmenting space.
simpler proof: set x_0 = sqrt(c)

$$k(x, x') = (\langle x, x' \rangle + c)^d = \sum_{i=0}^{m} \binom{d}{i} (\langle x, x' \rangle)^i c^{d-i}$$

Individual terms $(\langle x, x' \rangle)^i$ are dot products for some $\Phi_i(x)$.

Circle Example



Dual Perceptron

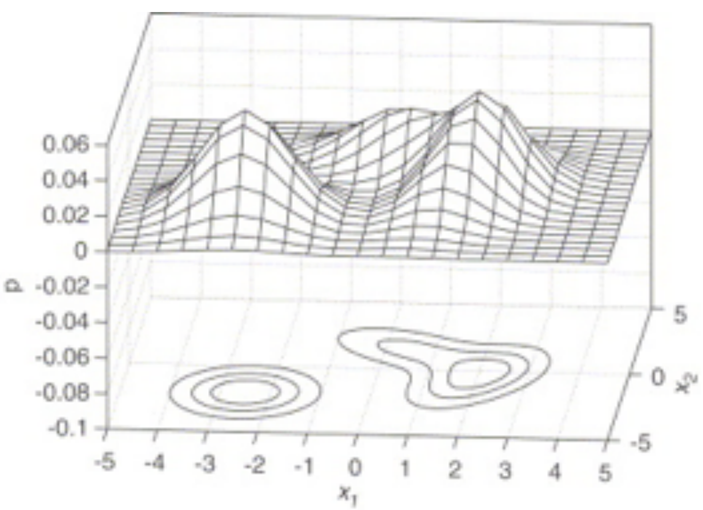
update on	new param.	w (implicit)
x1: +1	α = (+1, 0, 0, 0, 0)	ϕ(x1)
x2: -1	α = (+1, -1, 0, 0, 0)	ϕ(x1) - ϕ(x2)
x3: -1	α = (+1, -1, -1, 0, 0)	

$$k(x, x') = (x \cdot x')^2 \;\Leftrightarrow\; \phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2)$$

$$k(x, x') = (x \cdot x' + 1)^2 \;\Leftrightarrow\; \phi(x) = ?$$

Gaussian Kernels

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

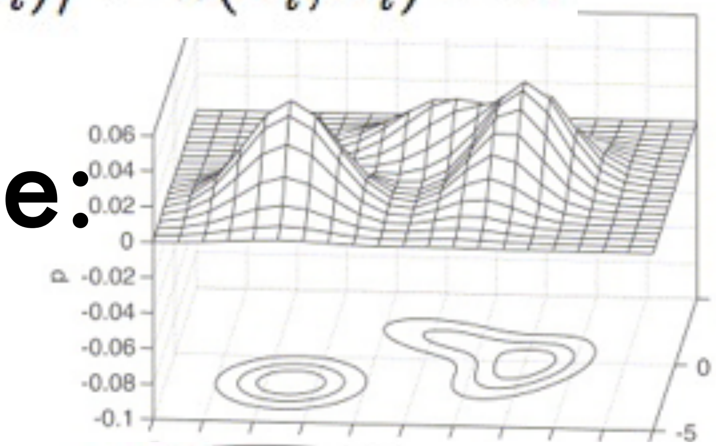


- distorts distance instead of angle (RBF kernels)
  - agreement with examples is now b/w 0 and 1
- geometric intuition in original space:
  - place a gaussian bump on each example
- geometric intuition in feature space (primal):
  - implicit mapping is N dimensional (N examples)
  - kernel matrix is full rank => independent bases

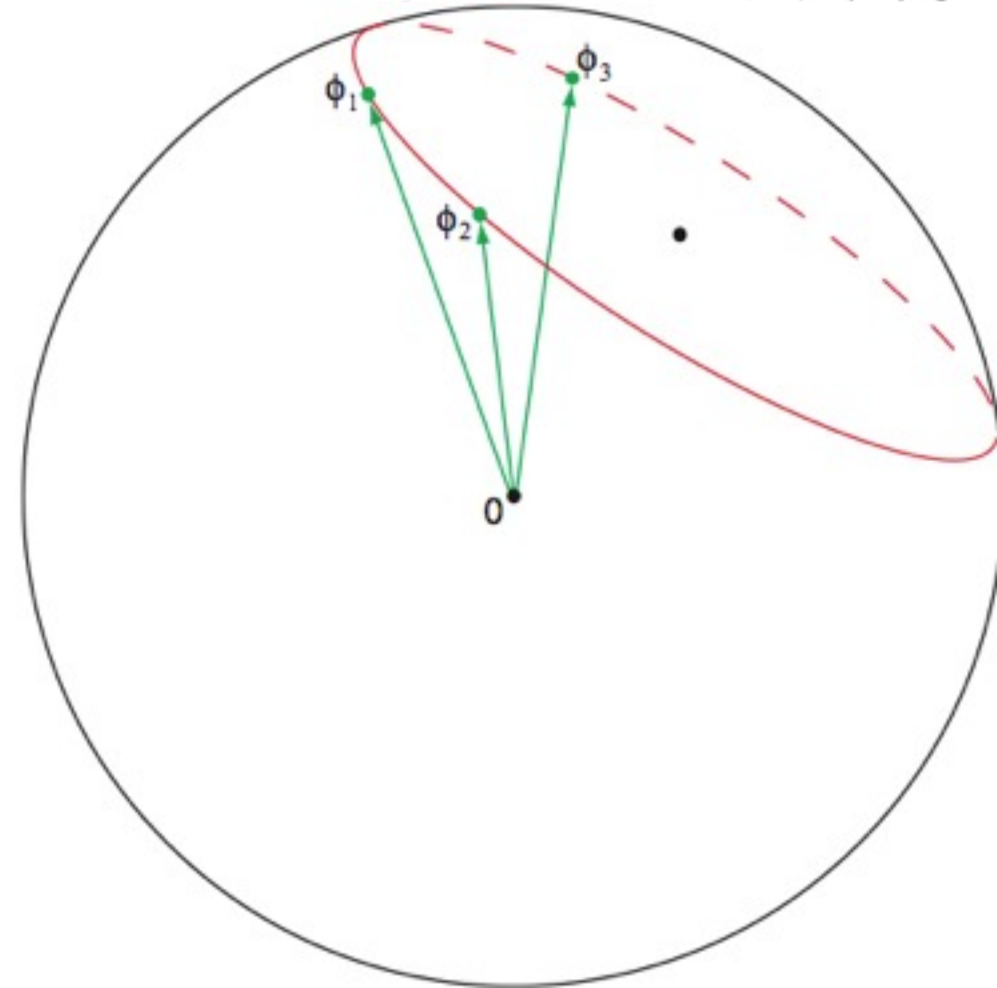
Gaussian Kernels

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

$$\|\phi(\mathbf{x}_i)\|^2 = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_i) \rangle = k(\mathbf{x}_i, \mathbf{x}_i) = 1.$$



- geometric intuition in feature space:
  - implicit mapping is N dimensional (N examples)
  - kernel matrix is full rank => independent bases
  - k(x,x) = 1 => all examples on unit hypersphere



Kernel Conditions

Computability

We have to be able to compute $k(x, x')$ efficiently (much cheaper than dot products themselves).

"Nice and Useful" Functions

The features themselves have to be useful for the learning problem at hand. Quite often this means smooth functions.

Symmetry

Obviously $k(x, x') = k(x', x)$ due to the symmetry of the dot product $\langle \Phi(x), \Phi(x') \rangle = \langle \Phi(x'), \Phi(x) \rangle$.

Dot Product in Feature Space

Is there always a Φ such that k really is a dot product?

Mercer's Theorem

The Theorem

For any symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which is square integrable in $\mathcal{X} \times \mathcal{X}$ and which satisfies

$$\int_{\mathcal{X} \times \mathcal{X}} k(x, x') f(x) f(x') dx dx' \geq 0 \text{ for all } f \in L_2(\mathcal{X})$$

there exist $\phi_i : \mathcal{X} \to \mathbb{R}$ and numbers $\lambda_i \geq 0$ where

$$k(x, x') = \sum_i \lambda_i \phi_i(x) \phi_i(x') \text{ for all } x, x' \in \mathcal{X}.$$

Interpretation

Double integral is the continuous version of a vector-matrix-vector multiplication. For positive semidefinite matrices we have

$$\sum \sum k(x_i, x_j) \alpha_i \alpha_j \geq 0$$

Properties

Distance in Feature Space

Distance between points in feature space via

$$
\begin{aligned}
d(x, x')^2 :=& \|\Phi(x) - \Phi(x')\|^2 \\
=& \langle \Phi(x), \Phi(x) \rangle - 2\langle \Phi(x), \Phi(x') \rangle + \langle \Phi(x'), \Phi(x') \rangle \\
=& k(x, x) + k(x', x') - 2k(x, x)
\end{aligned}
$$

Kernel Matrix

To compare observations we compute dot products, so we study the matrix K given by

$$K_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle = k(x_i, x_j)$$

where x_i are the training patterns.

Similarity Measure

The entries K_{ij} tell us the overlap between $\Phi(x_i)$ and $\Phi(x_j)$, so $k(x_i, x_j)$ is a similarity measure.

Properties

K is Positive Semidefinite

Claim: $\alpha^\top K \alpha \geq 0$ for all $\alpha \in \mathbb{R}^m$ and all kernel matrices $K \in \mathbb{R}^{m \times m}$. Proof:

$$\begin{aligned}\sum_{i,j}^m \alpha_i \alpha_j K_{ij} &= \sum_{i,j}^m \alpha_i \alpha_j \langle \Phi(x_i), \Phi(x_j) \rangle \\ &= \left\langle \sum_i^m \alpha_i \Phi(x_i), \sum_j^m \alpha_j \Phi(x_j) \right\rangle = \left\| \sum_{i=1}^m \alpha_i \Phi(x_i) \right\|^2\end{aligned}$$

Kernel Expansion

If w is given by a linear combination of $\Phi(x_i)$ we get

$$\langle w, \Phi(x) \rangle = \left\langle \sum_{i=1}^m \alpha_i \Phi(x_i), \Phi(x) \right\rangle = \sum_{i=1}^m \alpha_i k(x_i, x).$$

A Counterexample

A Candidate for a Kernel

$$k(x, x') = \begin{cases} 1 & \text{if } \|x - x'\| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

This is symmetric and gives us some information about the proximity of points, yet it is not a proper kernel ...

Kernel Matrix

We use three points, $x_1 = 1, x_2 = 2, x_3 = 3$ and compute the resulting "kernelmatrix" K. This yields

$$K = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$ and eigenvalues $(\sqrt{2}-1)^{-1}, 1$ and $(1-\sqrt{2})$.

as eigensystem. Hence k is not a kernel.

Examples

you only need to know polynomial and gaussian.

Examples of kernels $k(x, x')$

Linear	$\langle x, x' \rangle$
Laplacian RBF	$\exp\left(-\lambda \|x - x'\|\right)$
Gaussian RBF	$\exp\left(-\lambda \|x - x'\|^2\right)$
Polynomial	$\left(\langle x, x' \rangle + c\rangle\right)^d, c \geq 0, \ d \in \mathbb{N}$
B-Spline	$B_{2n+1}(x - x')$
Cond. Expectation	$\mathbf{E}_c[p(x\|c)p(x'\|c)]$

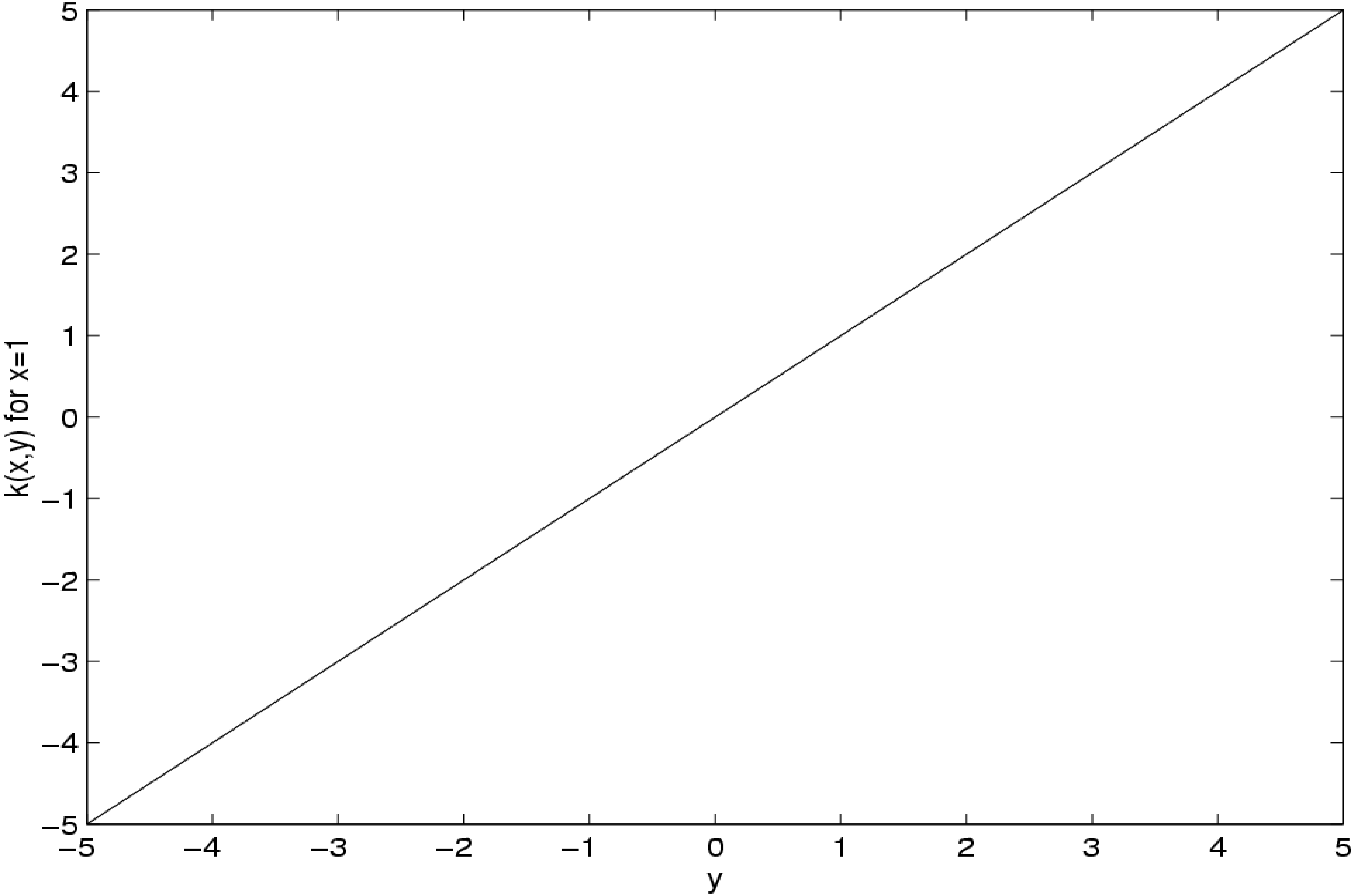
distorts distance

distorts angle

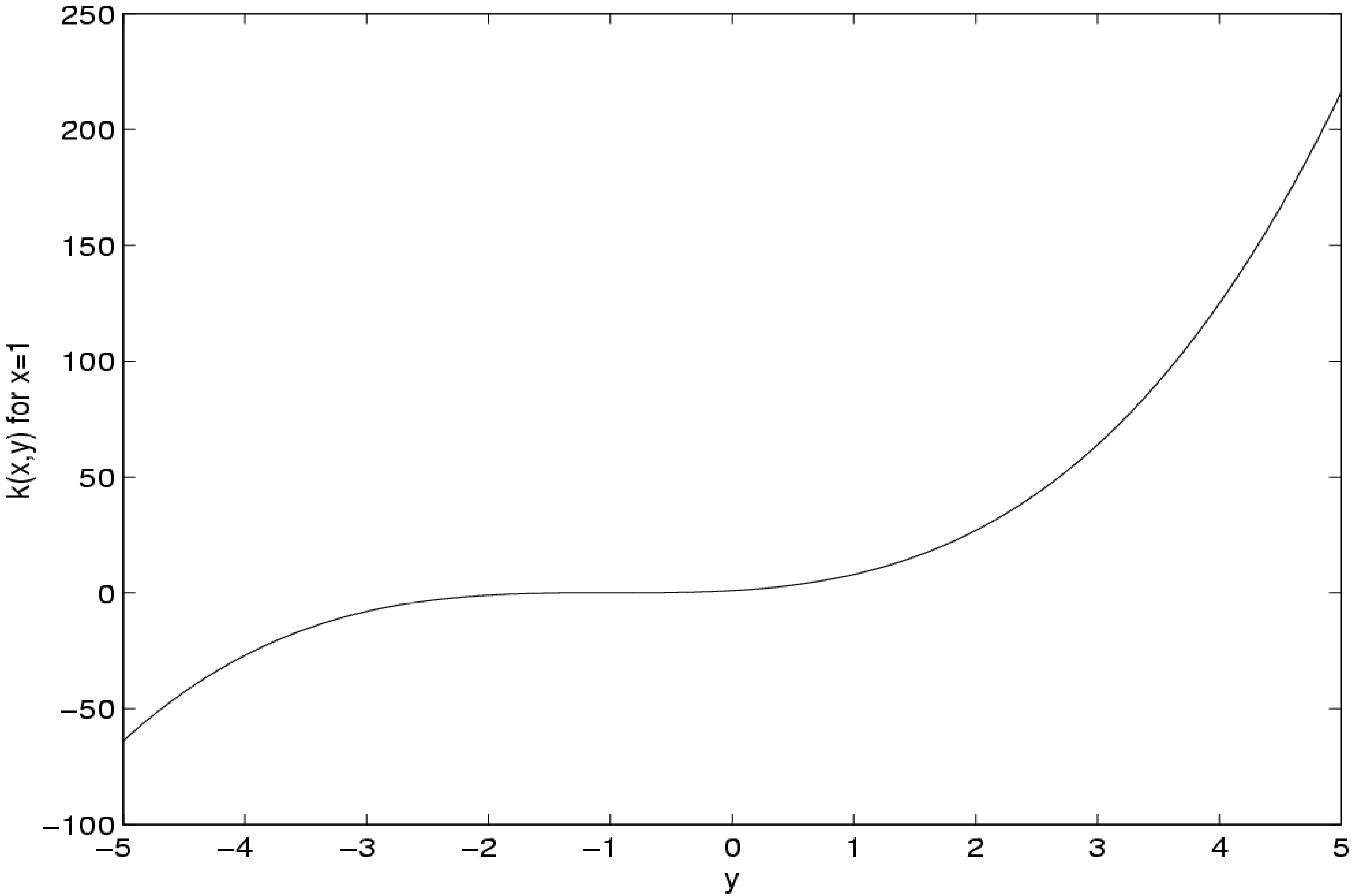
Simple trick for checking Mercer's condition

Compute the Fourier transform of the kernel and check that it is nonnegative.

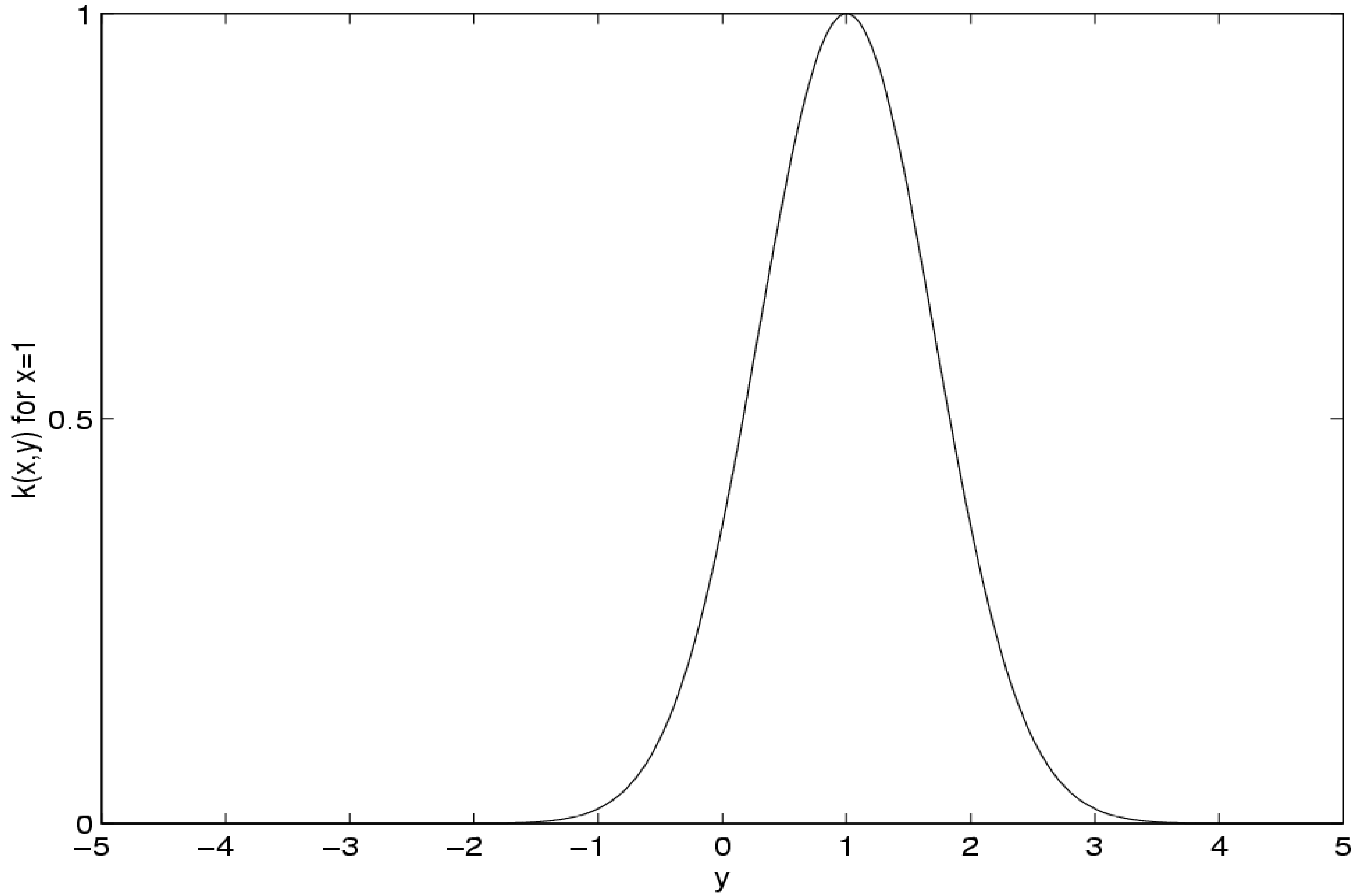
Linear Kernel



Polynomial of order 3



Gaussian Kernel



Summary

- Perceptron
  - Hebbian learning & biology
  - Algorithm
  - Convergence analysis
- Features and preprocessing
  - Nonlinear separation
  - Perceptron in feature space
- Kernels
  - Kernel trick
  - Properties
  - Examples