Support Vector Learning: Concepts and Algorithms

Alexander Johannes Smola
Department of Engineering and RSISE
Australian National University
Canberra, 0200 ACT, Australia
Alex.Smola@anu.edu.au

THE AUSTRALIAN
NATIONAL UNIVERSITY

Outline

What is Learning?

Classification, Regression, Novelty Detection

Linear Methods

Discriminant Analysis, Perceptron, Maximum Margin Classifier, Single Class Separation, LMS-Regression, SVM-Regression.

Preprocessing and Kernels

Standardization, Whitening, Feature Extraction, Kernels, Representer Theorem

Building Kernels

Algebraic Operations, Fourier Transforms, Taylor Expansions

Optimization for SVM

Chunking, Sequential Minimal Optimization, Online Methods

Baysics

Likelihood, Prior, and Posterior Probability, MAP estimation, Gaussian Processes.

Supervised Learning

Goal

We have some empirical data (images, medical observations, market indicators, socioeconomical background of a person, texts), say $\mathbf{x}_i \in \mathcal{X}$, usually together with labels $y_i \in \mathcal{Y}$ (digits, diseases, share price, credit rating, relevance) and we want to find a function that connects $\mathbf{x}$ with y.

Cost of Misprediction

Typically, there will be a function $c(\mathbf{x}, y, f(\mathbf{x}))$ depending on $\mathbf{x}, y$ and the prediction $f(\mathbf{x})$ which tells us the loss we incur by estimating $f(\mathbf{x})$ instead of the actual value y. This may be, e.g. the number of wrong characters in an OCR application, the cost for wrong treatment, the actual loss of money on the stock exchange, the cost for a bankrupt client, or the amount of annoyance by receiving a wrong e-mail.

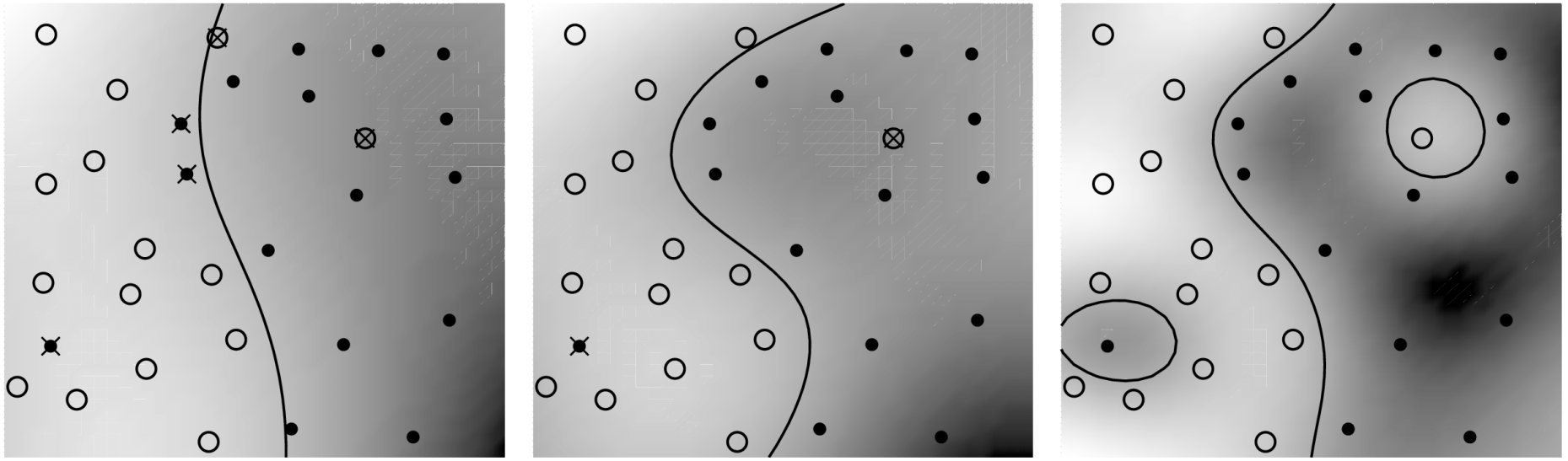
Classification

Goal

We want to find a function $f : \mathcal{X} \to \{\pm 1\}$ or $f : \mathcal{X} \to \mathbb{R}$ which will tell us whether a new observation $\mathbf{x} \in \mathcal{X}$ belongs to class 1 or -1 (and possibly the confidence with which this is so.

Questions

How to rate f, how to find f, how to interpret $f(\mathbf{x})$, ...



Applications

Optical Character Recognition

The goal is to classify (handwritten) characters (note that here f has to map into $\{a, \ldots, z\}$ rather than into $\{-1, 1\}$) automatically (form readers, scanners, post).

Spam Filtering

Determine whether an e-mail is spam or not (or whether it is urgent), based on keywords, word frequencies, special characters (\$, !, uppercase, whitespace), ...

Medical Diagnosis

Given some observations such as immune status, blood pressure, etc., determine whether a patient will develop a certain disease. Here it matters that we can estimate the probability of such an outcome.

Face Detection

Given a patch of an image, determine whether this corresponds to a face or not (useful for face tracking later).

Regression

Goal

We want to find a function $f : \mathcal{X} \to \mathbb{R}$ which will tell us the value of f at $\mathbf{x}$.

Application: Getting Rich

Predict the stock value of IBM/CISCO/BHP/TELSTRA . . . given today's market indicators (plus further background data).

Application: Wafer Fab

Predict (and optimize) the yield for a microprocessor, given the process parameters (temperature, chemicals, duration, . . .).

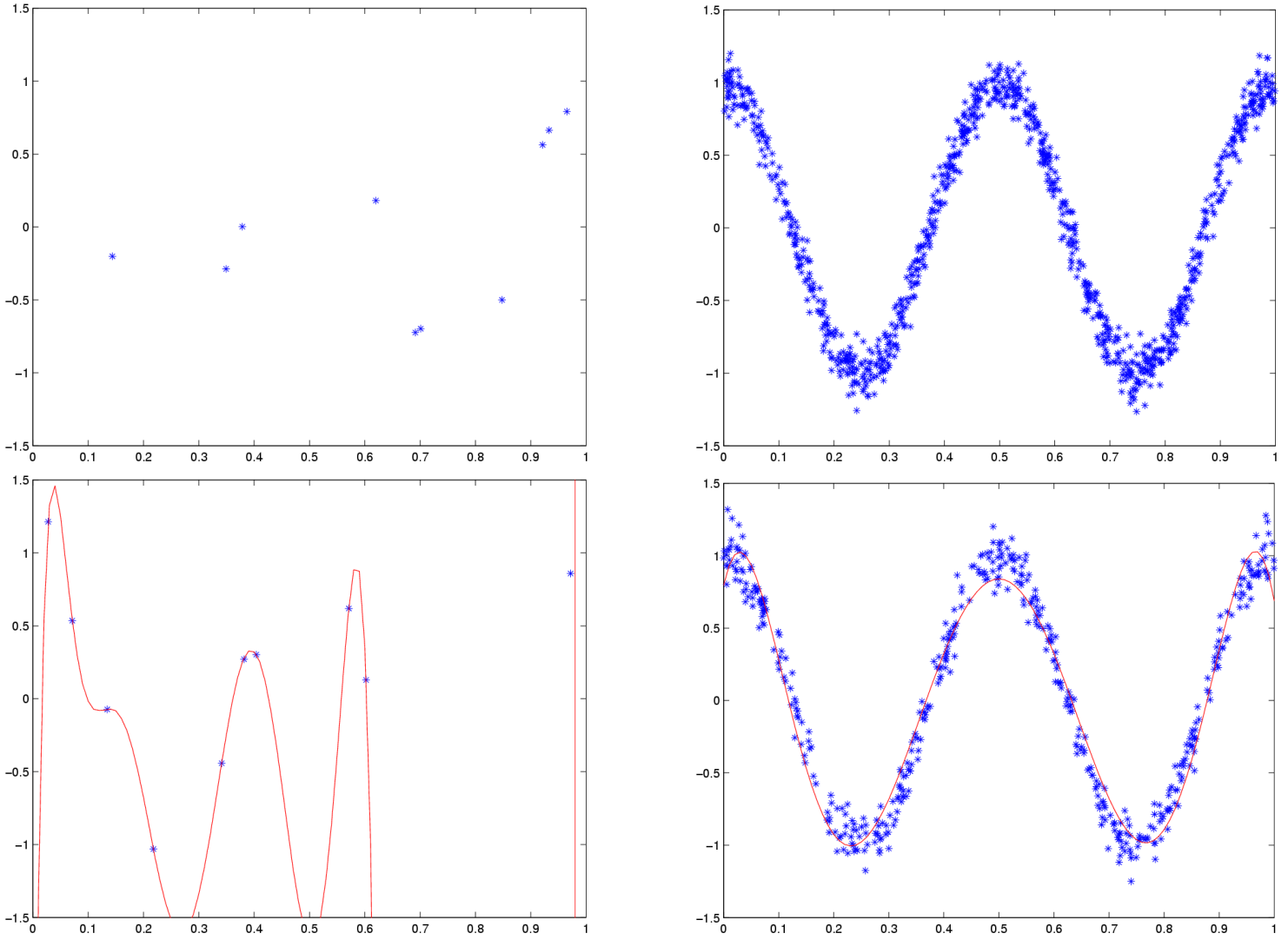
Application: Network Routers

Predict the network traffic through some hubs/routers/switches. We could reconfigure the infrastructure in time . . .

Application: Drug Release

Predict the requirement for a certain drug (e.g. insulin) automatically.

Example



Novelty Detection

Goal

- Build estimator that finds unusual observations and outliers
- Build estimator that can assess how typical an observation is

Idea

- Data is generated according to some density $p(x)$. Find regions of low density.
- Such areas can be approximated as the **level set** of an auxiliary function. No need to estimate $p(x)$ directly — use proxy of $p(x)$.
- Specifically: find $f(\mathbf{x})$ such that $\mathbf{x}$ is novel if $f(\mathbf{x}) \leq c$ where c is some constant, i.e. $f(\mathbf{x})$ describes the amount of novelty.

Applications

Network Intrusion Detection

Detect whether someone is trying to hack the network, downloading tons of MP3s, or doing anything else *unusual* on the network.

Jet Engine Failure Detection

You can't (normally) destroy a couple of jet engines just to see *how* they fail.

Database Cleaning

We want to find out whether someone stored bogus information in a database (typos, etc.), mislabelled digits, ugly digits, bad photographs in an electronic album.

Fraud Detection

Credit Card Companies, Telephone Bills, Medical Records

Self calibrating alarm devices

Car alarms (adjusts itself to where the car is parked), home alarm (location of furniture, temperature, open windows, etc.)

A Simple Pattern Recognition Algorithm

Learning Problem

Classify a set of observations $\mathbf{x}_i$ into the classes $y_i = 1$ and $y_i = -1$.

Simple Idea

Compute means of classes $y_i = 1$ and $y_i = -1$ by

$$\mathbf{c}_+ = \frac{1}{m_+} \sum_{\{i:y_i=+1\}} \mathbf{x}_i \text{ and } \mathbf{c}_- = \frac{1}{m_-} \sum_{\{i:y_i=-1\}} \mathbf{x}_i$$

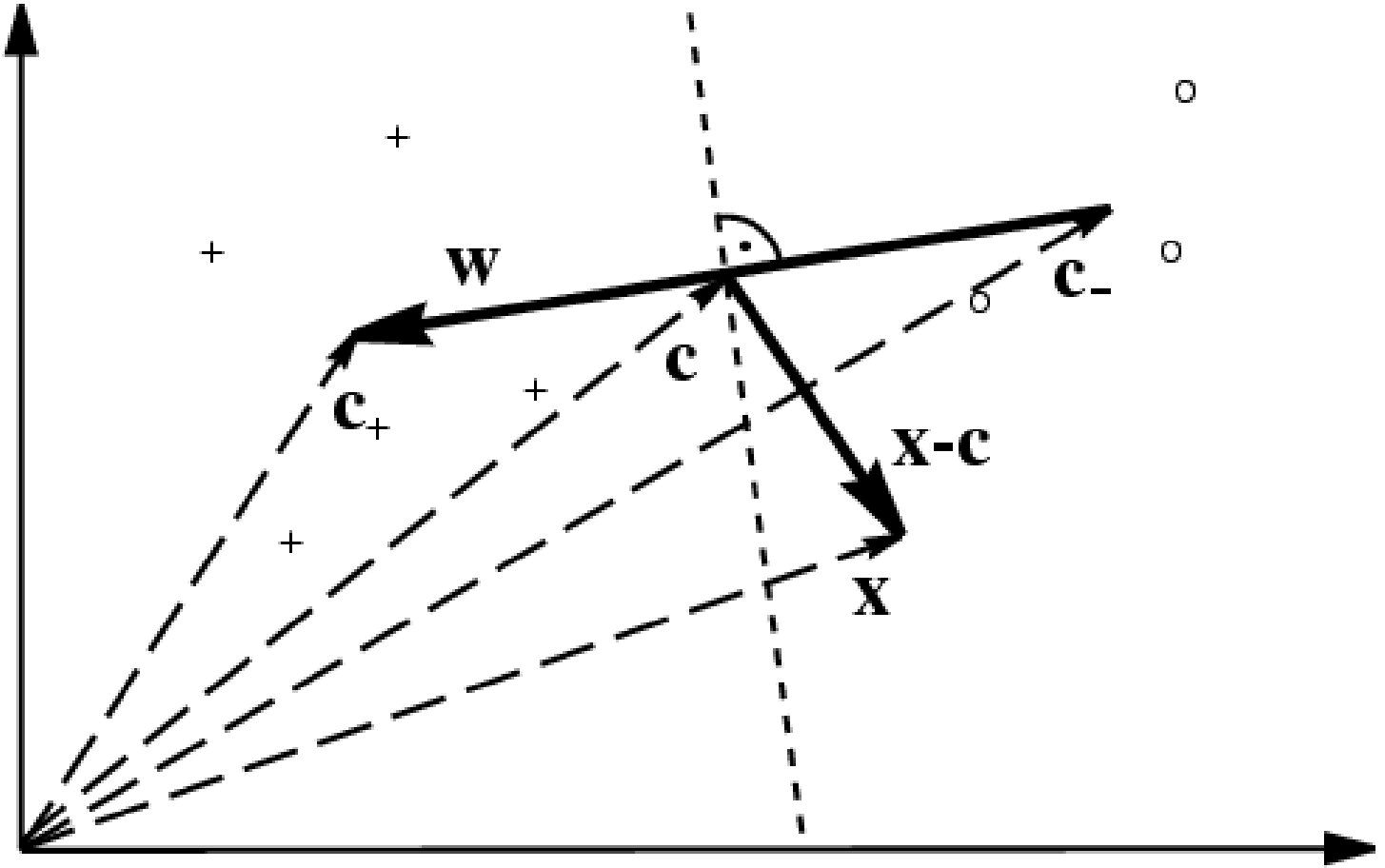
and classify a novel point according to which of the two class means $\mathbf{c}_+$ and $\mathbf{c}_-$ is closer. This leads to the following decision function:

$$f(\mathbf{x}) = \langle \mathbf{c}_+ - \mathbf{c}_-, \mathbf{x} \rangle - \left\langle \mathbf{c}_+ - \mathbf{c}_-, \frac{\mathbf{c}_+ + \mathbf{c}_-}{2} \right\rangle = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

where $\mathbf{w} = \mathbf{c}_+ - \mathbf{c}_-$ and $b = \frac{1}{2}(\|\mathbf{c}_-\|^2 - \|\mathbf{c}_+\|^2)$

We can compute $f(\mathbf{x})$ via dot products between $\mathbf{x}_i$ and $\mathbf{x}$.

A Simple Pattern Recognition Algorithm



The Perceptron

Problem

Somtimes, the means are not representative for the actual separation of the two classes $y_i = 1$ and $y_i = -1$.

Solution

Start with weight vector $\mathbf{w} = 0$ and threshold $b = 0$. Update $\mathbf{w}, b$ only if a mistake is made by adding some of the instances to $\mathbf{w}, b$. In other words, for mistakes

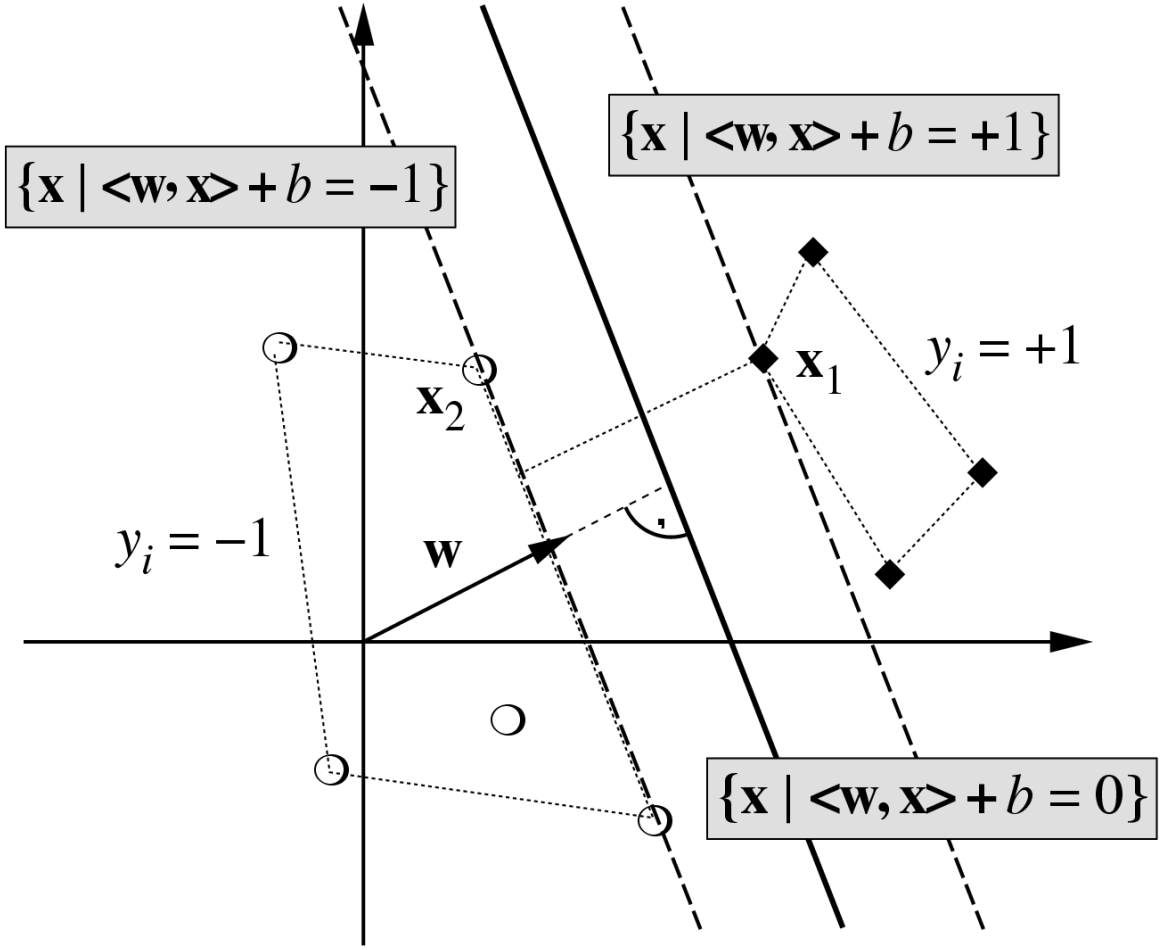
$$\mathbf{w} \longleftarrow \mathbf{w} + \eta y_i \mathbf{x}_i \text{ and } b \longleftarrow b + \eta y_i \text{ where } \eta > 0.$$

Convergence Theorem (Novikoff)

Suppose that there exists a $\rho > 0$, a weight vector $\mathbf{w}^*$ satisfying $\|\mathbf{w}^*\| = 1$, and a threshold b^* such that $y_i \left(\langle \mathbf{w}^*, \mathbf{x}_i \rangle + b^*\right) \geq \rho$ for all $1 \leq i \leq m$.

Then for all $\eta > 0$, the hypothesis maintained by the perceptron algorithm converges after no more than $(b^{*2} + 1)(R^2 + 1)/\rho^2$ updates, where $R = \max_i \|\mathbf{x}_i\|$. Clearly, the limiting hypothesis is consistent with the training data (X, Y).

Optimal Separating Hyperplane



Note:

$$\langle w, x_1\rangle + b = +1$$
$$\langle w, x_2\rangle + b = -1$$

$$\Rightarrow \quad \langle w, (x_1 - x_2)\rangle = 2$$

$$\Rightarrow \left\langle \frac{w}{\|w\|}, (x_1 - x_2) \right\rangle = \frac{2}{\|w\|}$$

Hard Margin

THE AUSTRALIAN NATIONAL UNIVERSITY

Goal

Find the hyperplane with maximum distance from both sets $y_i = 1$ and $y_i = -1$.

Linear Function

Hyperplanes are parametrized by linear functions $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$.

Hard Margin Version (no errors allowed)

Provided that we require

$$f(\mathbf{x}_i) \geq 1 \text{ if } y_i = 1 \text{ and } f(\mathbf{x}_i) \leq -1 \text{ otherwise,}$$

the margin between the two sets is given by $\frac{2}{\|\mathbf{w}\|}$.

Mathematical Programming Setting

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 \geq 0 \text{ for all } 1 \leq i \leq m \end{aligned}$$

Soft Margin

Problem

Often the data is noisy and we may not be able to find a separating hyperplane at all.

Solution (Bennet & Mangasarian 1992)

relax the constraints

$$y_i f(\mathbf{x}_i) \geq 1 \text{ into } y_i f(\mathbf{x}_i) \geq 1 - \xi_i$$

and add ξ_i into the objective function

Mathematical Programming Setting

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \xi_i \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 + \xi_i \geq 0 \text{ and } \xi_i \geq 0 \text{ for all } 1 \leq i \leq m \end{aligned}$$

We will show how to solve this problem later.

Regularized Risk Functionals

Rewriting the Optimization Problem

We can collapse the optimization problem into

$$\underset{\mathbf{w},b}{\text{minimize}} \frac{1}{2C}\|\mathbf{w}\|^2 + \sum_{i=1}^{m} \max(0, 1 - y_i f(\mathbf{x}_i))$$

$R_{\text{emp}}[f]$ Here the second term is the data-dependent expression. We could call it the **training error** or **empirical risk**.

$\Omega[f]$ The first term ensures that we restrict our choice of functions to simple functions, i.e., flat functions with small $\mathbf{w}$. It is also called the **regularization term**.

$R_{\text{reg}}[f]$ The sum of both terms is called the regularized risk functional and we have

$$R_{\text{reg}}[f] := R_{\text{emp}}[f] + \lambda\Omega[f]$$

Here $\lambda > 0$ is the regularization constant.

Least-Mean-Squares Regression

Learning Problem

Observations $\mathbf{x}_i$ together with target values y_i. Find the linear function $f(\mathbf{x})$ which minimizes the squared error of misprediction $(f(\mathbf{x}_i) - y_i)^2$.

Solution

Minimize the expression

$$\sum_{i=1}^{m}(f(\mathbf{x}_i) - y_i)^2 = \sum_{i=1}^{m}(\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i)^2 = \|\mathbf{y} - X\mathbf{w}\|^2.$$

Here we define $\mathbf{y} = (y_1, \ldots, y_m)$ and $X = [\mathbf{x}_1, \ldots, \mathbf{x}_m]$.

The minimizer can be obtained by setting the derivative to 0. This yields

$$\mathbf{w} = (X^\top X)^{-1} X^\top \mathbf{y}$$

Here again, $\mathbf{w}$ is a linear combination of the observations $\mathbf{x}_i$.

Regularized Least-Mean-Squares Regression

Problem

If we have too few data or $\mathbf{x}$ is too high dimensional, $X^\top X$ may not have full rank. This leads to instabilities.

Solution

Solve the setting as a regularized risk functional. Here we treat $\sum_i (f(\mathbf{x}_i) - y_i)^2$ as the empirical error term and $\frac{1}{2}\|\mathbf{w}\|^2$ as the regularization term. This leads to

$$R_{\text{reg}}[f] = \sum_{i=1}^{m} (f(\mathbf{x}_i) - y_i)^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2.$$

Optimal Parameters

Taking the derivative of $R_{\text{reg}}[f]$ with respect to $\mathbf{w}$ yields

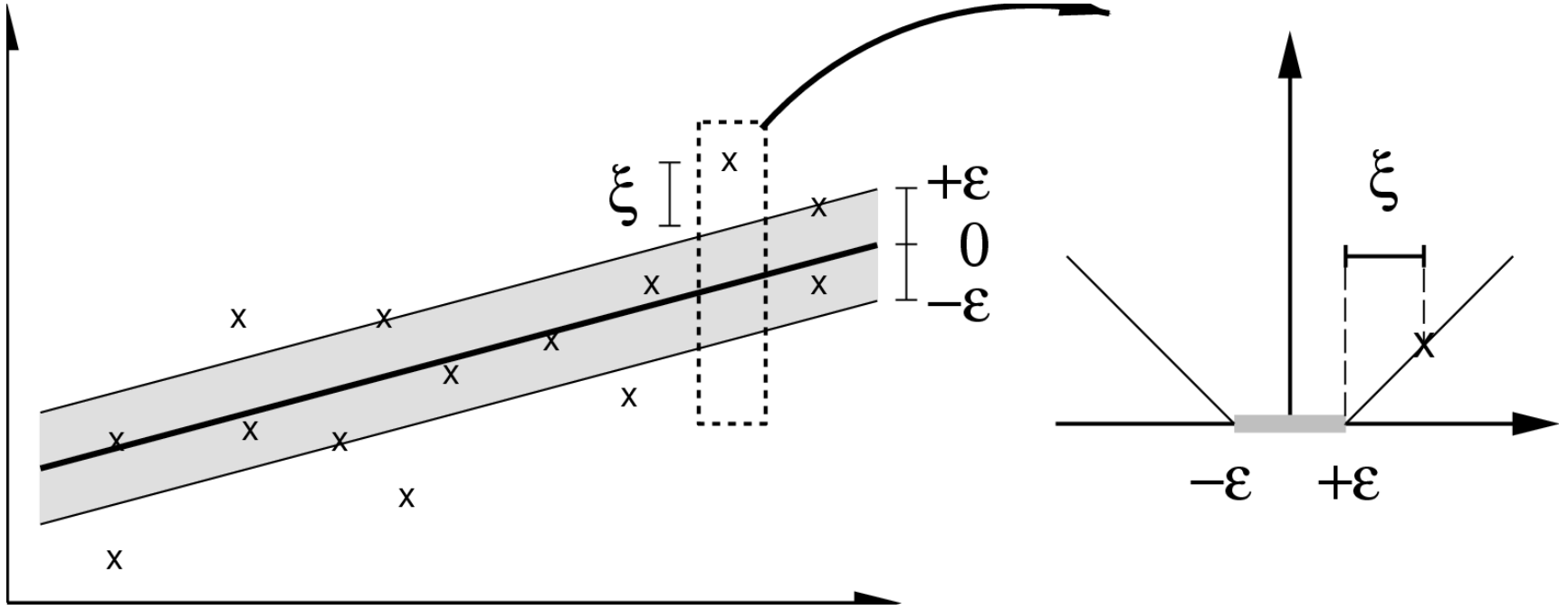
$$\mathbf{w} = (x^\top X + \lambda \mathbf{1})^{-1} X^\top \mathbf{y}.$$

This means that we add to the main diagonal of $X^\top X$ to make the optimization problem better conditioned.

ε-insensitive Linear Regressor

Optimization Problem

- $\frac{1}{2}\|\mathbf{w}\|^2$ subject to $-\varepsilon \leq y_i - (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \leq \varepsilon$
- $\frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{m} \xi_i + \xi_i^*$ subject to $-(\varepsilon + \xi_i) \leq y_i - (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \leq \varepsilon + \xi_i^*$



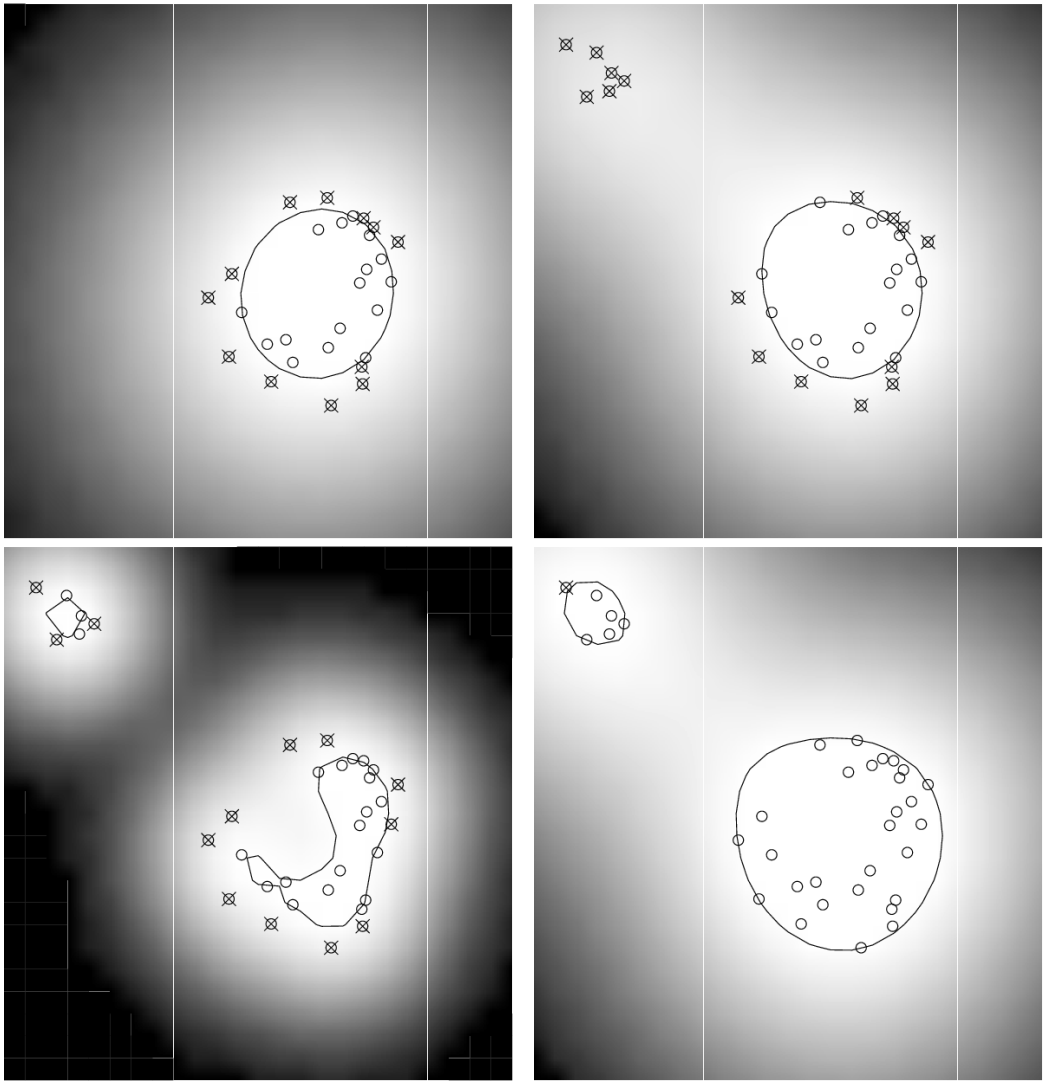
Novelty Detection

Data

Observations $\mathbf{x}_i$ generated from some $\Pr(\mathbf{x})$, e.g.,
(network usage patterns)
(handwritten digits)
(alarm sensors)
(factory status)

Task

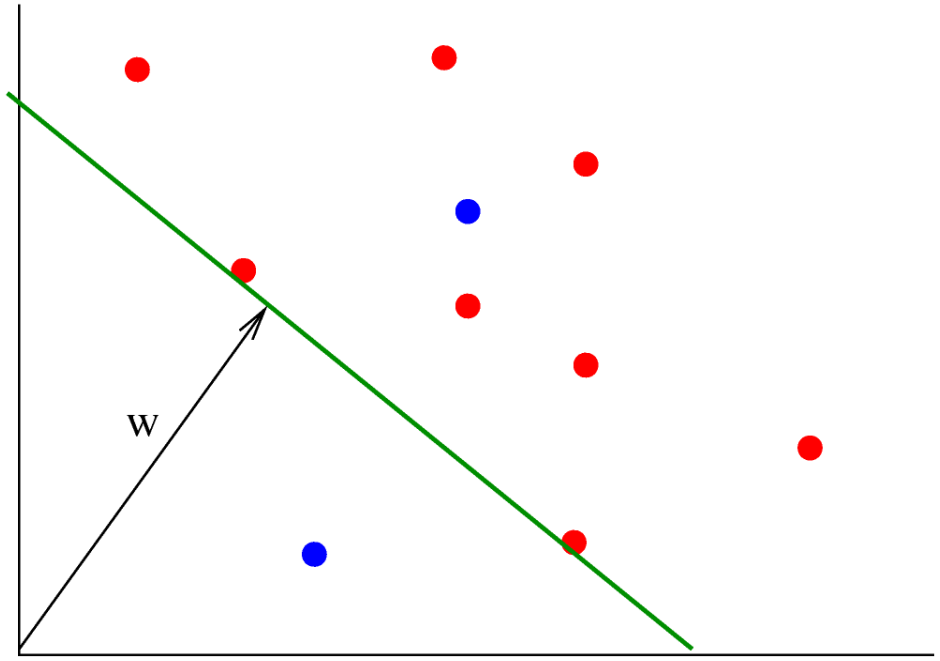
Find unusual events, clean database, distinguish typical examples.



Maximum Distance Hyperplane

THE AUSTRALIAN NATIONAL UNIVERSITY

Idea Find hyperplane, parameterized by $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = 0$ that has **maximum distance from origin** yet is still closer to the origin than the observations.



Hard Margin

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1 \end{aligned}$$

Soft Margin

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \xi_i \\ \text{subject to} \quad & \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned}$$

The Dual Optimization Problem

Primal Problem

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{subject to} \quad & \langle \mathbf{w}, \mathbf{x}_i \rangle - 1 + \xi_i \geq 0 \text{ and } \xi_i \geq 0 \end{aligned}$$

Lagrange Function

We construct a **Lagrange Function** L by subtracting the constraints, multiplied by **Lagrange multipliers** (α_i and η_i), from the **Primal Objective Function**. L has a **saddlepoint** at the optimal solution.

$$L = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle - 1 + \xi_i\right) - \sum_{i=1}^{m}\eta_i\xi_i \text{ where } \alpha_i, \eta_i \geq 0$$

For instance, if $\xi_i < 0$ we could increase L without bound via η_i.

The Dual Optimization Problem II

Optimality Conditions

$$\partial_{\mathbf{w}} L = \mathbf{w} - \sum_{i=1}^{m} \alpha_i \mathbf{x}_i = 0 \implies \mathbf{w} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i$$
$$\partial_{\xi_i} L = C - \alpha_i - \eta_i = 0 \implies \alpha_i \in [0, C]$$

Now we **substitute** the two optimality conditions **back into** L.

Dual Problem

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \sum_{i=1}^{m} \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \sum_{i=1}^{m} \alpha_i \\ \text{subject to} \quad & \alpha_i \in [0, C] \end{aligned}$$

All this is only possible due to the convexity of the primal problem.

The ν-Trick

Problem

Depending on how we choose C, the number of points selected as lying on the "wrong" side of the hyperplane $H := \{\mathbf{x}|\langle \mathbf{w}, \mathbf{x}\rangle = 1\}$ will vary.

But we would like to **specify a certain fraction** ν beforehand.

Solution

Use adaptive hyperplane that separates data from the origin, i.e. find $H := \{\mathbf{x}|\langle \mathbf{w}, \mathbf{x}\rangle = \rho\}$ where the threshold ρ is **adaptive**.

Primal Problem

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{m} \xi_i - m\nu\rho$$

$$\text{subject to} \quad \langle \mathbf{w}, \mathbf{x}_i\rangle - \rho + \xi_i \geq 0 \text{ and } \xi_i \geq 0$$

Dual Problem

$$\text{minimize} \quad \frac{1}{2}\sum_{i=1}^{m} \alpha_i\alpha_j\langle \mathbf{x}_i, \mathbf{x}_j\rangle$$

$$\text{subject to} \quad \alpha_i \in [0,1] \text{ and } \sum_{i=1}^{m} \alpha_i = \nu m.$$

Optimization Problems with the ν-Trick

THE AUSTRALIAN NATIONAL UNIVERSITY

Classification

$$\text{minimize} \quad \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle$$

$$\text{subject to} \quad \alpha_i \in [0,1], \sum_{i=1}^{m}\alpha_i = \nu m \text{ and } \sum_{i=1}^{m}\alpha_i y_i = 0.$$

$$\text{function expansion} \quad f(\mathbf{x}) = \sum_{i=1}^{m}\alpha_i y_i \langle \mathbf{x}_i, \mathbf{x}\rangle + b$$

Regression

$$\text{minimize} \quad \frac{1}{2}\sum_{i,j=1}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle \mathbf{x}_i, \mathbf{x}_j\rangle - \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i^*)$$

$$\text{subject to} \quad \alpha_i, \alpha_i^* \in [0, C] \text{ and } \sum_{i=1}^{m}(\alpha_i + \alpha_i^*) = C\nu m.$$

$$\text{function expansion} \quad f(\mathbf{x}) = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)\langle \mathbf{x}_i, \mathbf{x}\rangle + b$$

Preprocessing

Zero Mean

Different offsets may skew the estimates considerably. So it is convenient to transform

$$\mathbf{x} \rightarrow \Phi(\mathbf{x}) = \mathbf{x} - \text{Mean}(\mathbf{x})$$

Unit Variance

Sometimes, whitening of the variables helps. Then we use

$$\mathbf{x} \rightarrow \Phi(\mathbf{x}) = C^{-\frac{1}{2}} \left(\mathbf{x} - \text{Mean}(\mathbf{x})\right)$$

which leads to zero mean and unit covariance matrix. ICA would be another option, leading to decorrelated input variables.

Prior Knowledge

If we know that certain features $\Phi_i(\mathbf{x})$ are useful for our task, we expand $\mathbf{x}$ into

$$\mathbf{x} \rightarrow \Phi(\mathbf{x}) = (\mathbf{x}, \Phi_1(\mathbf{x}), \ldots, \Phi_n(\mathbf{x}))$$

Kernels and Nonlinearity

Insight 1 With the transformations $\Phi(\mathbf{x})$ we immediately obtain **nonlinear functions** $f(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + b$.

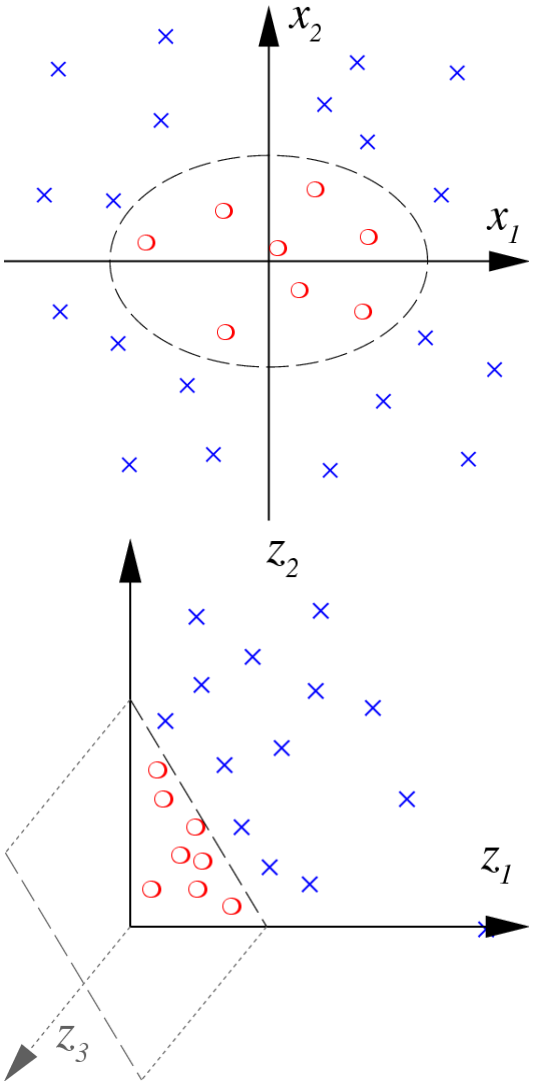
Insight 2 Since the data only appears in the form of dot products all we need to do is replace every $\langle x, x' \rangle$ by $\langle \Phi(x), \Phi(x') \rangle$.

This works for an arbitrary nonlinearity $\Phi(\mathbf{x})$.

Insight 3 instead of computing $\Phi(\mathbf{x})$ explicitly use **kernel function**

$$k(\mathbf{x}, \mathbf{x}') := \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle.$$

Strategy Replace every $\langle \mathbf{x}, \mathbf{x}' \rangle$ by $k(\mathbf{x}, \mathbf{x}')$.



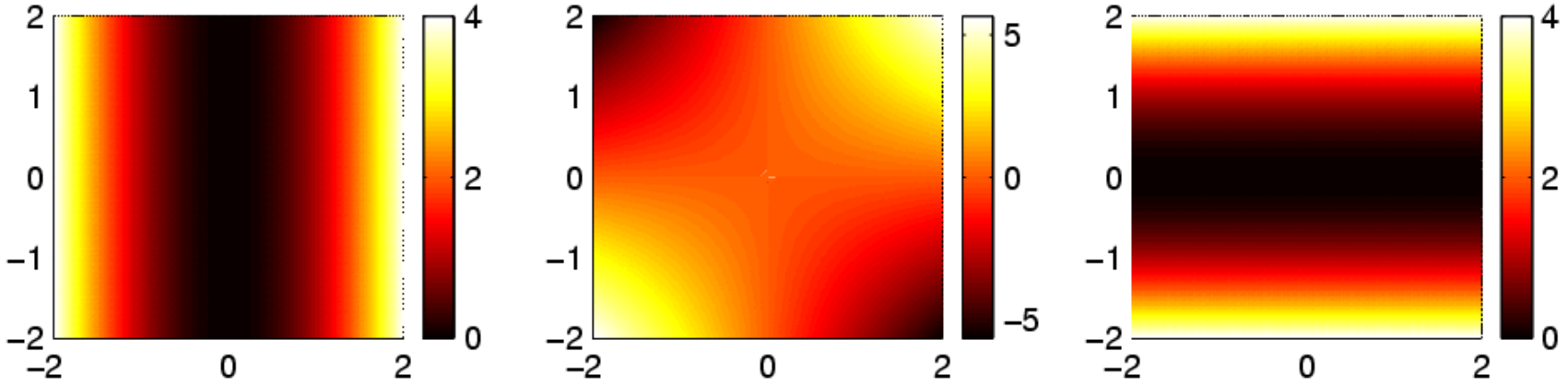
Example: Polynomial Kernels

Quadratic Features in $\mathbb{R}^2$ $\Phi(x) := \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$

Dot Product

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

Extension This dot product trick does not only work for 2nd order polynomials but for any order: $k(x, x') = \langle x, x' \rangle^d$.



Kernels and Mercer's Theorem

THE AUSTRALIAN NATIONAL UNIVERSITY

Question which functions $k(x, x')$ can we use as kernels?

Short Answer all functions $k(x, x')$ generating **symmetric** matrices $K \in \mathbb{R}^{m \times m}$ with $K_{ij} := k(x_i, x_j)$ where **all eigenvalues** are **nonnegative**.

Long Answer all $k(x, x')$ satisfying the conditions of **Mercer's Theorem**:

Any $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that satisfies

$$\int k(x, x') f(x) f(x') dx dx' \geq 0$$

for all $f \in L_2(\mathcal{X})$ can be written as

$$k(x, x') = \sum_i \lambda_i \phi_i(x) \phi_i(x')$$

where $\phi_i(x)$ are the eigenfunctions of the integral operator

The Representer Theorem

Question is there always a kernel expansion

$$f(x) = \sum_i \alpha_i k(x_i, x)$$

if we minimize

$$R_{\text{reg}}[w] = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} c(x_i, y_i, f(x_i))$$

Answer (Representer Theorem) **YES**

Kimeldorf and Wahba, 1971; Cox and O'Sulivan, 1990; This holds for SVM, Gaussian Processes, Regularization Networks, ...

Consequences the number of kernels **increases** with the number of observations m. Finding the optimal solutions will typically cost $O(m^2)$ to $O(m^3)$ operations (depending on the data).

Support Vector Classification with Kernels

Primal Optimization Problem this is identical to the non-kernelized one, only that $\mathbf{x}$ is replaced by $\Phi(\mathbf{x})$.

Dual Optimization Problem

$$\text{minimize} \quad \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{subject to} \quad \alpha_i \in [0,1], \sum_{i=1}^{m} \alpha_i = \nu m \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0.$$

$$\text{function expansion} \quad f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b$$

Note the size of the optimization problem has not increased — we still have to deal with m variables, regardless of the dimensionality of $\mathbf{w}$.

Regression and Novelty Detection

THE AUSTRALIAN NATIONAL UNIVERSITY

Regression

$$\text{minimize} \quad \frac{1}{2}\sum_{i,j=1}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i^*)$$

$$\text{subject to} \quad \alpha_i, \alpha_i^* \in [0, C] \text{ and } \sum_{i=1}^{m}(\alpha_i + \alpha_i^*) = C\nu m.$$

$$\text{function expansion} \quad f(\mathbf{x}) = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)k(\mathbf{x}_i, \mathbf{x}) + b$$

Novelty Detection

$$\text{minimize} \quad \frac{1}{2}\sum_{i=1}^{m}\alpha_i\alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{subject to} \quad \alpha_i \in [0, 1] \text{ and } \sum_{i=1}^{m}\alpha_i = \nu m.$$

$$\text{function expansion} \quad f(\mathbf{x}) = \sum_{i=1}^{m}\alpha_i k(\mathbf{x}_i, \mathbf{x})$$

Zoo: Translation Invariant Kernels

Theorem: Any **translation invariant symmetric** $k(x, x') = k(x - x')$ satisfies Mercer's condition if it has a **nonnegative Fourier transform**.

The **decay properties of the spectrum** tell us the **smoothness** of k.

Gaussian RBF Kernels $$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

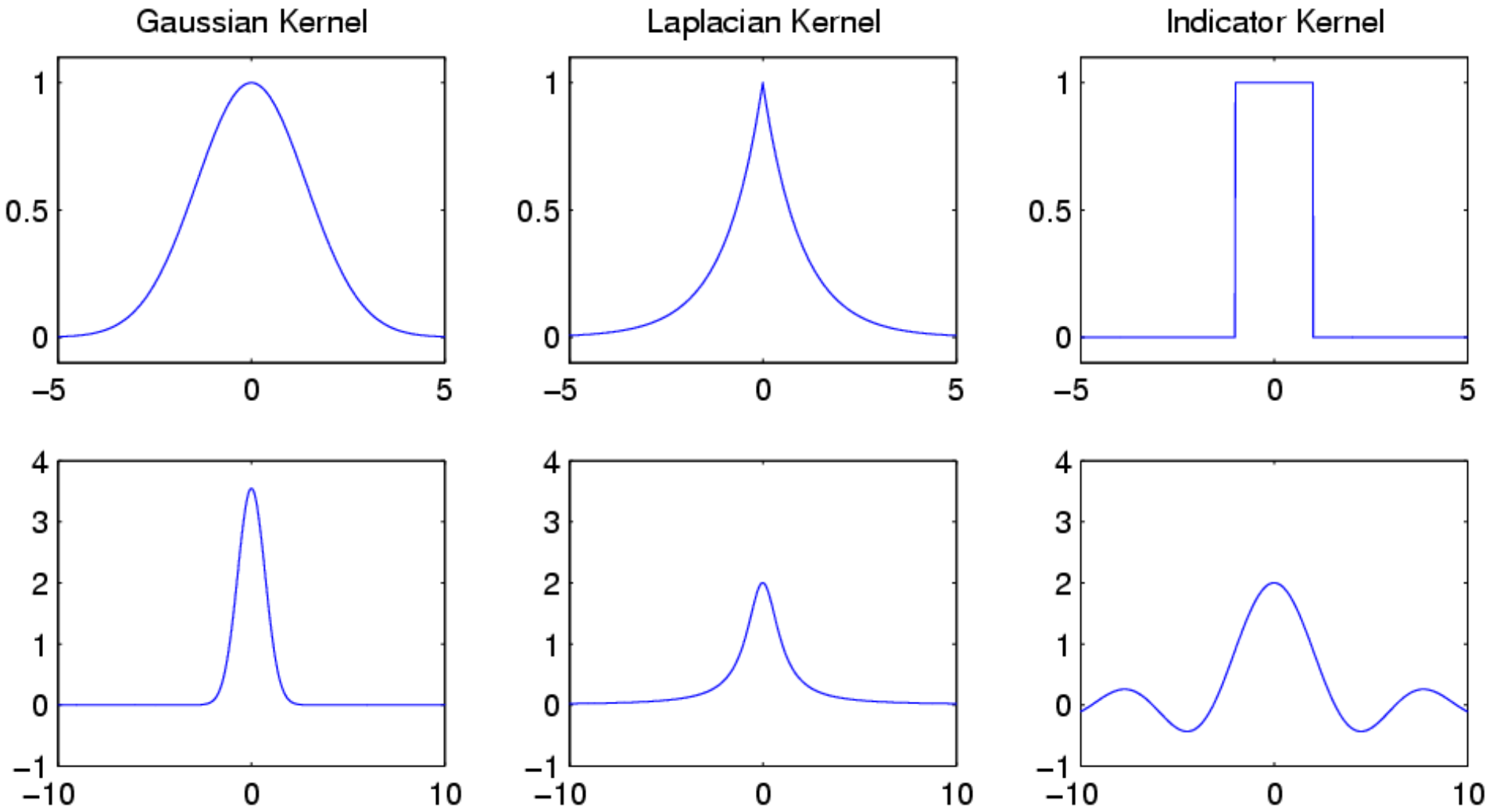
The Fourier transform is also a Gaussian, however with inverse width.

Laplacian RBF Kernels $$k(x, x') = \exp\left(-\frac{\|x - x'\|}{\sigma}\right)$$

The Fourier spectrum decays less rapidly (damped harmonic oscillator in one dimension).

Indicator Function $k(x, x') = \mathbf{1}_{[-1,1]}(x - x')$ is not a Mercer kernel since its Fourier transform, the sinc function, has negative entries.

Zoo: Translation Inveriant Kernels



Zoo: Dot Product Kernels

Theorem Any **dot product** kernel $k(x, x') = k(\langle x, x' \rangle)$ satisfies Mercer's condition if it has a **nonnegative Taylor series expansion**:

$$k(\mathbf{x}, \mathbf{x}') = \sum_{i=0}^{\infty} a_i \langle \mathbf{x}, \mathbf{x}' \rangle^i$$

The **decay of the Taylor expansion** determines the **smoothness** of k.

Polynomial Kernels $k(x, x') = (\langle x, x' \rangle + c)^d$

This is a proper kernel only for $d \in \mathbb{N}$ and $c \geq 0$.

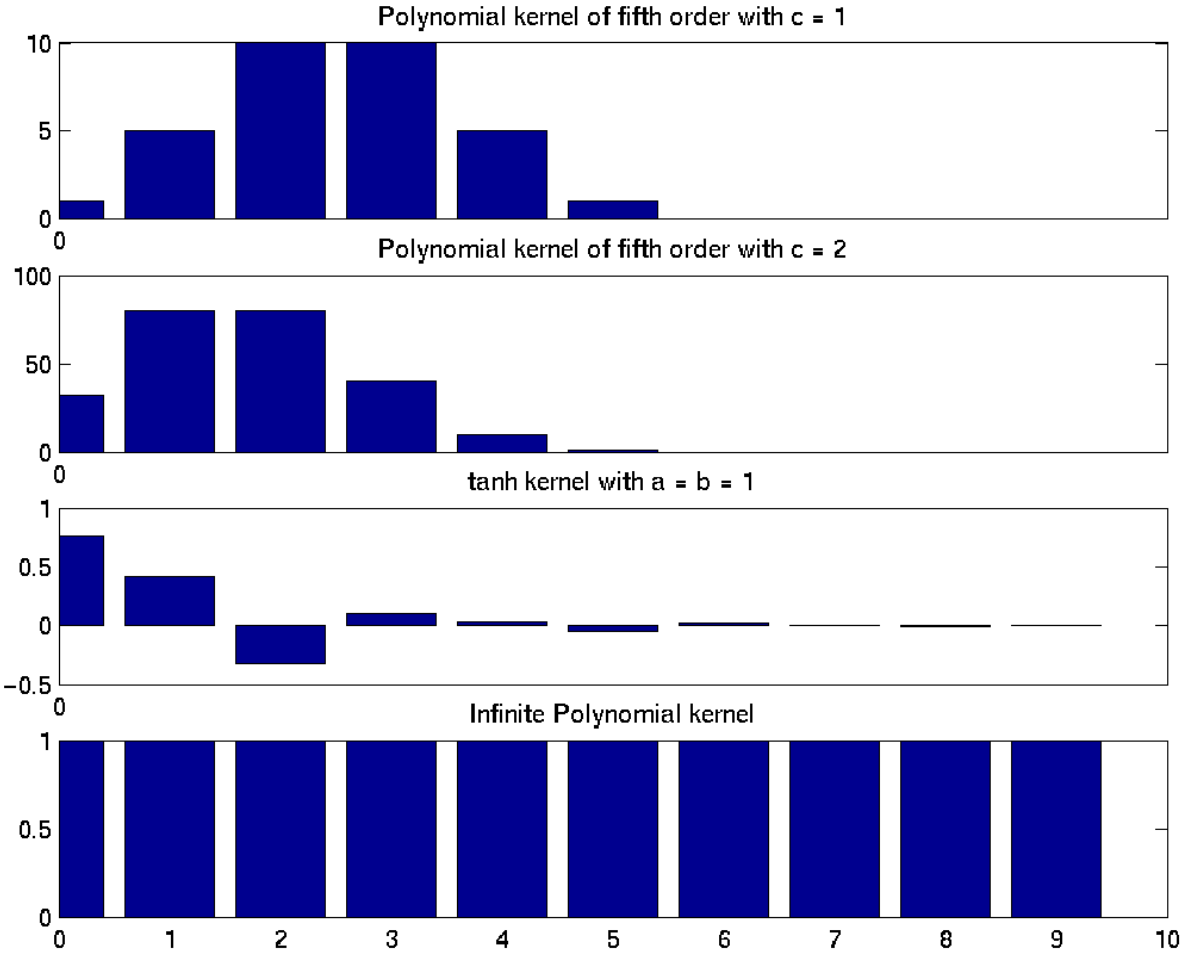
tanh-Kernels

$k(x, x') = \tanh(a\langle x, x' \rangle + b)$ is *never* a proper kernel, for any $a, b \in \mathbb{R}$.

Infinite Polynomials $k(x, x') = \dfrac{1}{1 - \langle x, x' \rangle}$ for $\|x\|, \|x'\| \leq 1$

is a Mercer kernel with bad regularization properties (flat Taylor series).

Zoo: Dot Product Kernels



Zoo: Making Even More Kernels

THE AUSTRALIAN NATIONAL UNIVERSITY

Sums for two Mercer kernels k_1 and k_2, k is a kernel if

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$$

Products $k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') \cdot k_2(\mathbf{x}, \mathbf{x}')$

Integrals for an arbitrary function $\kappa(\mathbf{x}, \mathbf{x}')$, k is a kernel if

$$k(\mathbf{x}, \mathbf{x}') = \int \kappa(\mathbf{x}, \mathbf{z})\kappa(\mathbf{x}', \mathbf{z})d\mathbf{z}$$

For instance, use a conditional probability distribution $\kappa(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})$ (Watkins' conditional independence kernel).

Explicit Maps for an arbitrary function $\Phi(\mathbf{x})$ and a positive definite matrix M

$$k(\mathbf{x}, \mathbf{x}') = \Phi^\top(\mathbf{x})M\Phi(\mathbf{x})$$

is a kernel. For instance, choose M to be the inverse Fisher information matrix and $\Phi(\mathbf{x})$ the Fisher scores (Haussler & Jaakkola).

How to write a Paper with Kernels

1. Take old and trusty linear algorithm from the 60s
2. Write it in the form of dot products
3. Replace dot products by kernel function
4. Make sure the complexity of the estimate does not go overboard (large $\|\mathbf{w}\|$). If necessary, add regularization
5. Perform theoretical analysis.
6. Write paper.

Newton Method

We want to minimize $f(\mathbf{x})$. Use quadratic approximation and solve at each step for the minimum of the latter explicitly. We get $f'(\mathbf{x}) + f''(\mathbf{x})\boldsymbol{\epsilon} = 0$ which yields the following algorithm:

Require: $\mathbf{x}_0$, Precision ϵ

Set $\mathbf{x} = \mathbf{x}_0$

repeat

$\mathbf{x} = \mathbf{x} - \frac{f'(\mathbf{x})}{f''(\mathbf{x})}$

until $|f'(\mathbf{x})| \leq \epsilon$

Output: $\mathbf{x}$

Convergence of Newton Method Let $g : \mathbb{R} \to \mathbb{R}$ be a twice continuously differentiable function and denote by $\mathbf{x}^* \in \mathbb{R}$ a point with $f''(\mathbf{x}^*) \neq 0$ and $f'(\mathbf{x}^*) = 0$. Then, provided $\mathbf{x}_0$ is sufficiently close to $\mathbf{x}^*$, the sequence generated by the Newton method will converge to $\mathbf{x}^*$ at least quadratically.

Convex Optimization Techniques

Interior Point Methods

solve constraints for primal and dual optimization problem simultaneously and satisfy the Kuhn-Tucker conditions. This involves inverting K, i.e. $O(m^3)$.

Sherman-Morrison-Woodbury Methods

exploit the fact that K does not really have full rank and invert a **rank-n approximation** of K. This leads to $O(m \cdot n^2)$ algorithms.

Chunking and Sequential Minimal Optimization

take subproblems of the big quadratic minimization problem and optimize over a subset variables at a time. Smaller memory footprint and sometimes fast.

Online Methods

stochastic gradient descent on the objective function. Simple, **approximate** algorithm, small memory footprint.

Flashback: Singleclass Dual QP

Dual Quadratic Program for ν-SVM

$$\text{minimize} \quad \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j K_{ij}$$

$$\text{subject to} \quad \alpha_i \in [0,1] \text{ and } \sum_{i=1}^{m}\alpha_i = \nu m.$$

Splitting divide the m patterns into a working set S^w and a fixed set S_f. Minimizing the objective function over S^w can only decrease it.

Subproblem on S_w

$$\text{minimize} \quad \frac{1}{2}\sum_{i,j\in S_w}^{m}\alpha_i^w\alpha_j^w K_{ij}^{ww} + \sum_{i\in S^w, j\in S^f}\alpha_i^w\alpha_j^f K_{ij}^{wf}$$

$$\text{subject to} \quad \alpha_i^w \in [0,1] \text{ and } \sum_{i\in S^w}^{m}\alpha_i^w = \nu m - \sum_{i\in S^f}\alpha_i^f.$$

Subset Selection and Chunking

Simple Version

Start with small set, train, keep SVs, add in new patterns, repeat. This works great for almost noise free data, e.g. USPS and NIST OCR problems (Vapnik & Chervonenkis, 1965, AT&T OCR Group, 1990es).

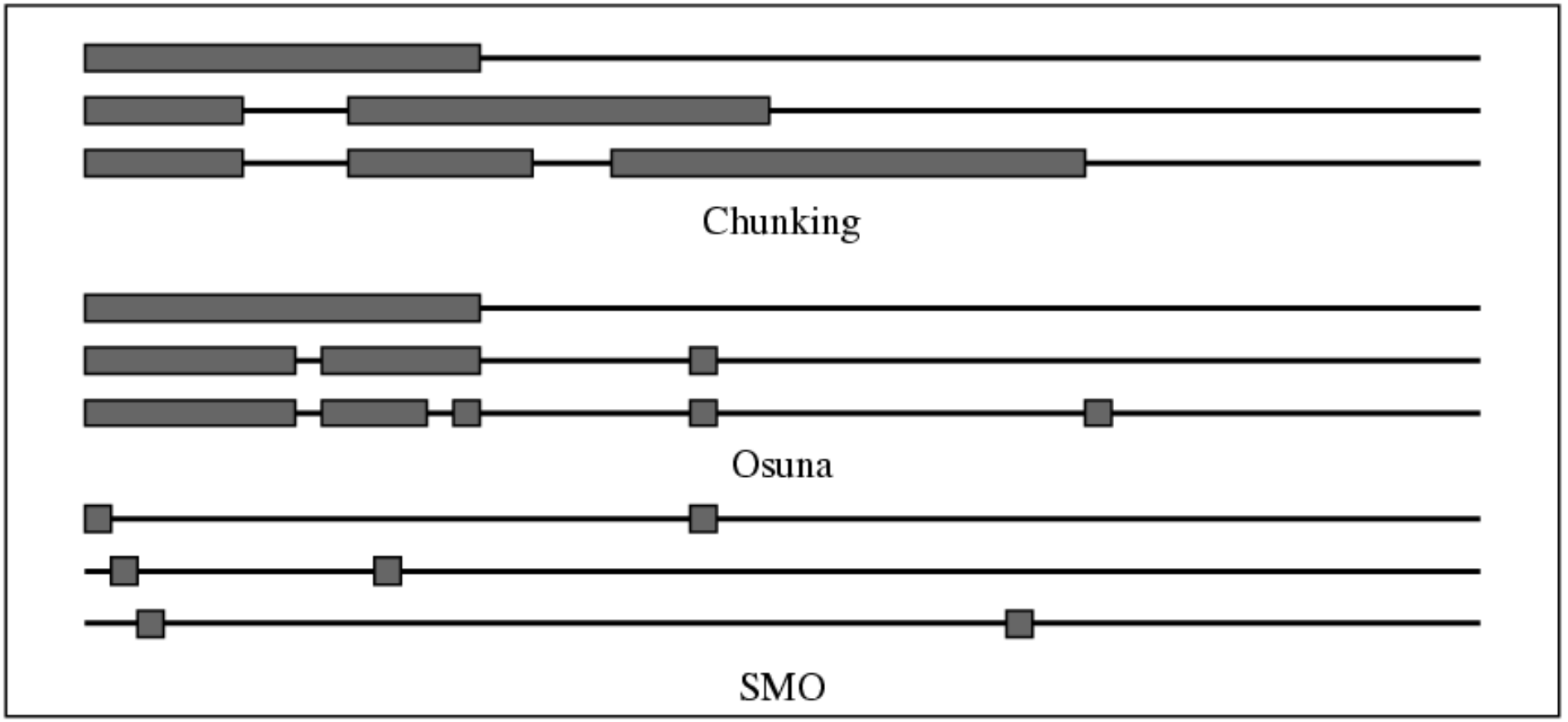
Basic Idea

Take subset of variables, say $\alpha_i \ldots \alpha_I$, optimize over the latter while keeping the rest fixed. Then pick next set of variables, optimize, ... (Osuna, Freund, Girosi, 1997). Probably best implementation by Joachims (SVMLight). Works great on texts.

Common Problems

Chunking does not always converge in practice and speed of convergence is **highly problem dependent**. Which variables should you pick? Performance degrades with the number of additional linear constraints. Most problems occur if many variables are neither 0 nor C.

Chunking Strategies



Sequential Minimal Optimization

Basic Idea

Optimize only over pairs of variables — we need pairs to keep the equality constraints satisfied (Platt, 1998).

Advantage

Analytic solution of subproblems is possible (at least for linear and quadratic loss functions), simple one-dimensional convex minimization otherwise.

Scaling Behaviour

Very large problems can be solved at only $O(m)$ storage cost, provided we are willing to wait long enough (time scales with $O(m^\gamma)$ where $\gamma > 2$, depending on the choice of problems and modifications).

Problems

Some formulations are hard to deal with in SMO: many nonzero start variables, several constraints at the same time (as in ν-SVM), special selection strategy for most modifications.

Explicit Solution for 2 Variables

Dual Quadratic Program in 2 Variables

$$\text{minimize} \quad \frac{1}{2}\sum_{i,j=1}^{2}\alpha_i\alpha_j K_{ij} + \sum_{i=1}^{2}\alpha_i v_i$$

$$\text{subject to} \quad \alpha_i \in [0,1] \text{ and } \sum_{i=1}^{2}\alpha_i = \Delta$$

Here $\Delta = \nu m - \sum_{i=3}^{m}\alpha_i$ and $C_i = \sum_{j=3}^{m}\alpha_j K_{ij}$.

Reduce Problem to 1 Variable

Use $\alpha_1 = \Delta - \alpha_2$ and substitute into the restricted optimization problem. This is a quadratic function on an interval, hence we can find the solution $\alpha_2^{\text{new}} = \dfrac{\Delta(K_{11} - K_{12}) - C_1 + C_2}{K_{11} + K_{22} - 2K_{12}}$ and have to restrict it such that α_2^{new} and $\Delta - \alpha_2^{\text{new}}$ are in $[0,1]$.

Selection Strategy

Problem

which variables i, j should we select?

Idea 1

we only make progress if α_i is not already satisfying the optimality conditions, i.e. $f(\mathbf{x}_i) > \rho$ for $\alpha_i = 0$ or $f(\mathbf{x}_i) < \rho$ for $\alpha_i = 1$.

Idea 2

we can show that the change by optimizing over i, j depends on $\frac{C_i - C_j}{K_{11} + K_{22} - 2K_{12}}$.

Strategy, Part 1

find i where C_i is large and where simultaneously α_i does not satisfy the constraints.

Strategy, Part 2

find j where $C_i - C_j$ is large and where also α_j does not satisfy the constraints.

Stopping Rule

Problem

when have we optimized enough?

Idea 1

use a lower bound for the objective function and stop if the relative gap size is small enough.

Idea 2

recall that the KKT-conditions tell us the gap size (i.e., the terms we subtracted in the Lagrange function).

Gap for Novelty Detection

$$\text{Gap} = \sum_{i=1}^{m} \alpha_i \max(f(\mathbf{x}_i) - \rho, 0) + (C - \alpha_i) \max(\rho - f(\mathbf{x}_i), 0)$$

This also makes a good selection criterion for optimization over subsets in other chunking strategies. Similar methods can be used for classification and regression.

Online Learning: Motivation

THE AUSTRALIAN NATIONAL UNIVERSITY

Problem:

Training complexity increases with sample size and number of basis functions. Typically training time scales with $O(m^{2+\gamma})$ and prediction time with $O(m)$.

Problem:

Distributions may change over time and we want to have a time dependent predictor.

Problem:

Iterative reduced set methods using projection are too expensive.

Design Goal:

Mainly **local** observations should matter for a predictor. Limited time horizon

Design Goal: **Cheap and simple** update rules for added patterns.

Design Goal: Almost as good as batch learning.

Online Learning: HOWTO

THE AUSTRALIAN NATIONAL UNIVERSITY

Start with regularized risk functional

$$R_{\text{reg}}[f] = \mathbf{E}\left[c(\mathbf{x}, y, f(\mathbf{x}))\right] + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

in the Reproducing Kernel Hilbert Space.

Perform stochastic gradient descent

on $R_{\text{reg}}[f]$, i.e. at step t (with $\mathbf{x}_t, y_t$) replace

$$\mathbf{E}\left[c(\mathbf{x}, y, f(\mathbf{x}))\right] \approx c(\mathbf{x}_t, y_t, f(\mathbf{x}_t)).$$

and walk λ in the negative gradient direction $\mathbf{w} \longrightarrow \mathbf{w} - \Lambda \partial_{\mathbf{w}} R_{\text{stoch}}[\mathbf{w}, t]$.

Stochastic Gradient:

$$\partial_{\mathbf{w}} \left[c(\mathbf{x}_t, y_t, f(\mathbf{x}_t)) + \frac{\lambda}{2}\|\mathbf{w}\|^2\right] = c'(\mathbf{x}_t, y_t, f(\mathbf{x}_t))\mathbf{x}_t + \lambda\mathbf{w}.$$

Update Rule

Update in Function Space:

$$\mathbf{w} \to \mathbf{w} - \lambda \partial_{\mathbf{w}} R_{\text{stoch}}[f, t] = (1 - \Lambda\lambda)\mathbf{w} - \Lambda c'(\mathbf{x}_t, y_t, f(\mathbf{x}_t))k(\mathbf{x}_t, \cdot).$$

Update in Coefficient Space:

$$\begin{aligned} \alpha_t &= -\Lambda c'(\mathbf{x}_t, y_t, f(\mathbf{x}_t)) \\ \alpha_i &= (1 - \Lambda\lambda)\alpha_i \text{ for } i < t \end{aligned}$$

We assume a kernel expansion $f(x) = \sum_{i=1}^{t} \alpha_i k(\mathbf{x}_i, \mathbf{x})$

Finite Time Horizon:

coefficients decay over time. After t iterations we have

$$\alpha_i \to (1 - \lambda\Lambda)^t \alpha_i$$

We can drop α_i after t steps with an error of at most $(1 - \lambda\Lambda)^t \sqrt{k(\mathbf{x}_i, \mathbf{x}_i)}$.

More about this tomorrow ...

Prior Probability

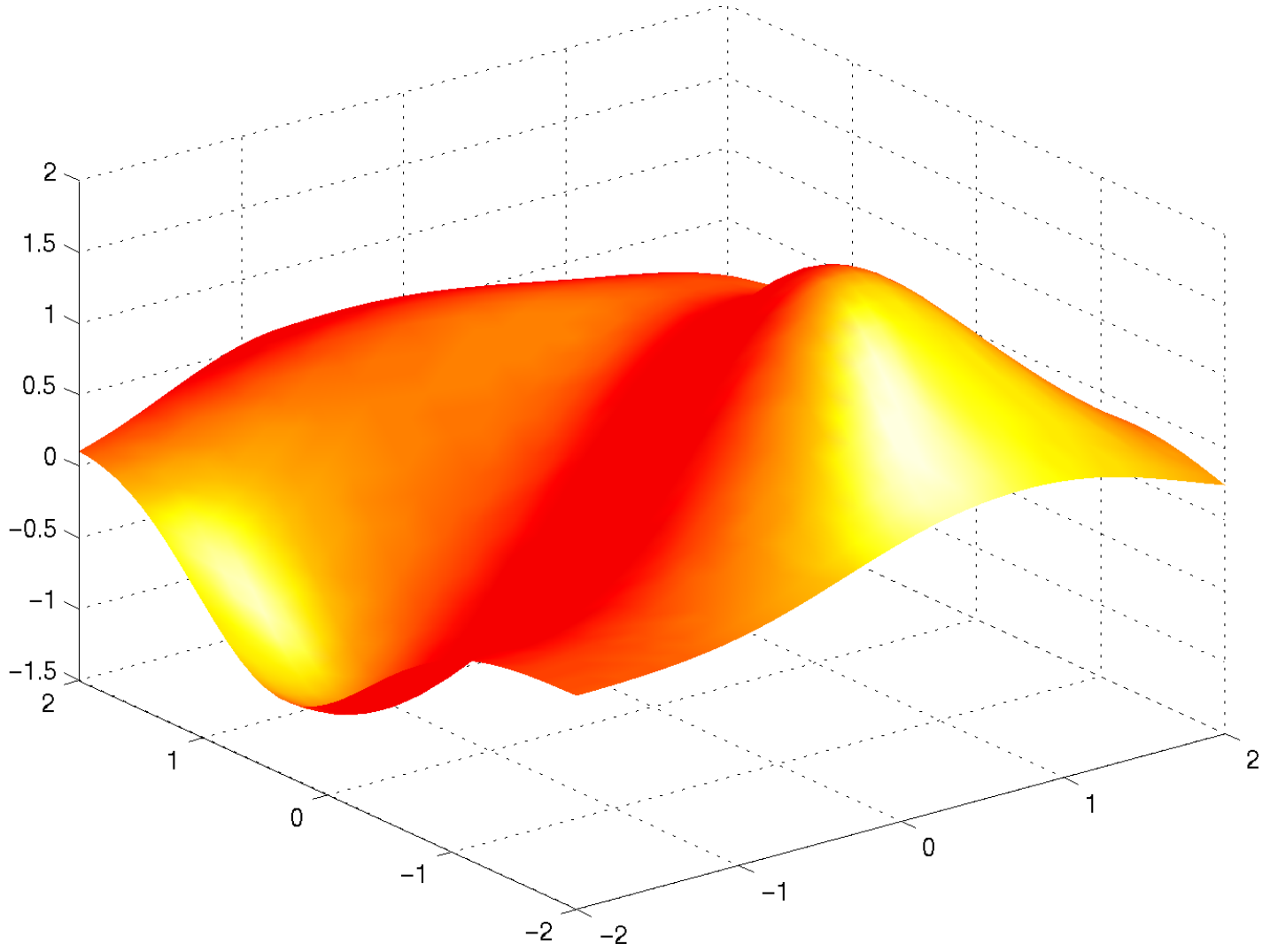
Idea 1

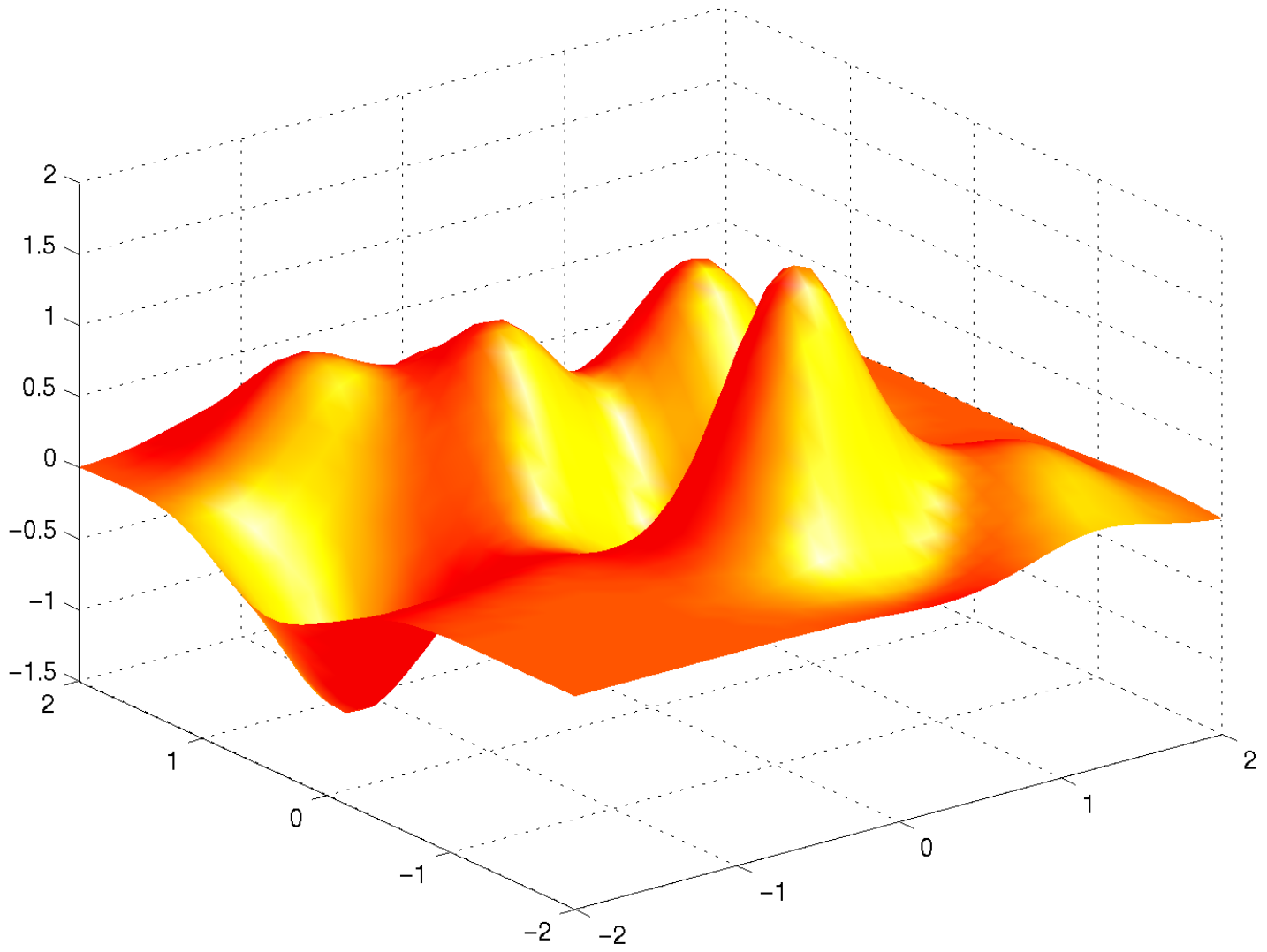
Quite often we have a rough idea of what function we can expect beforehand.

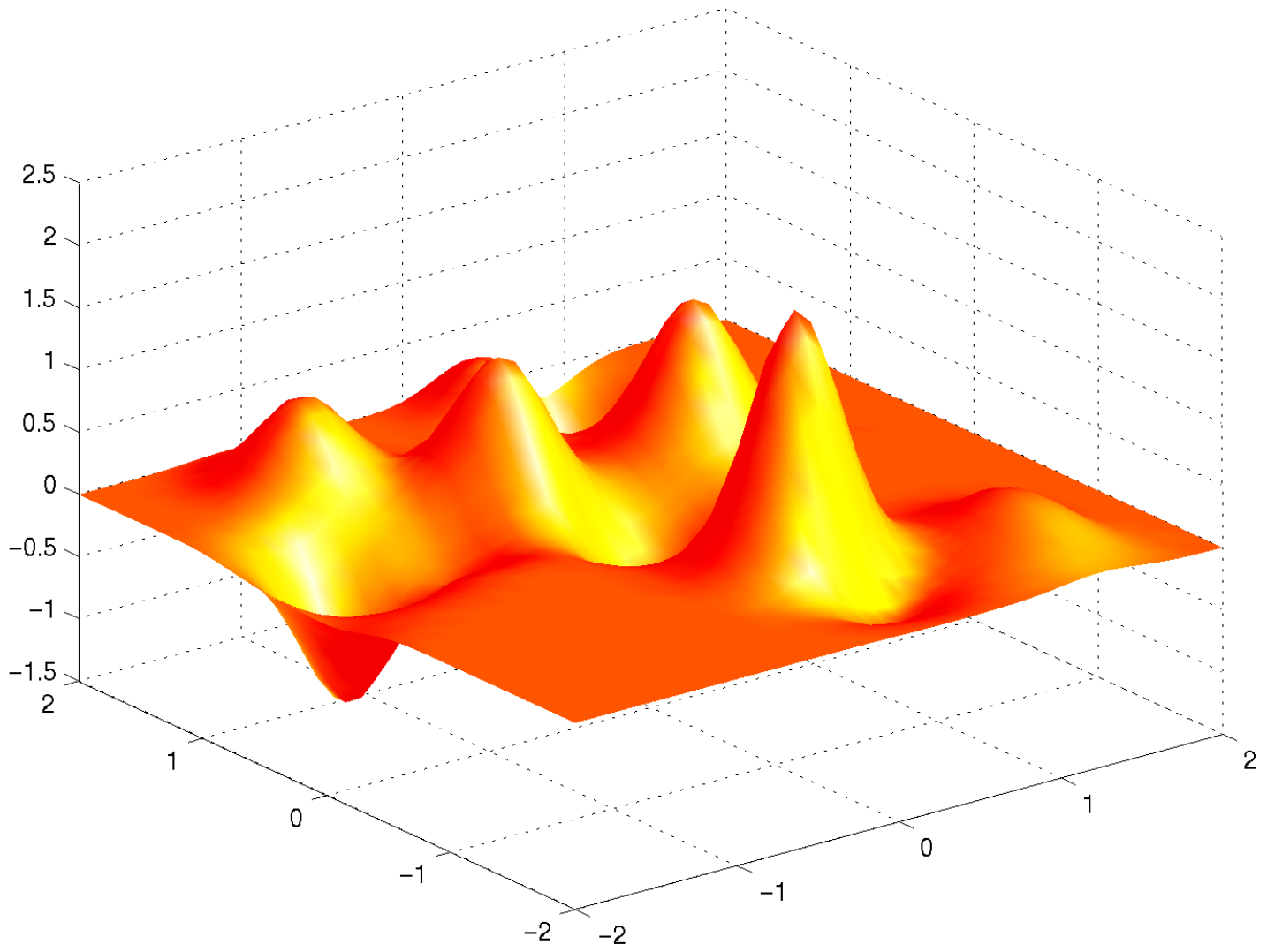
- We observe similar functions in practice.
- We **think** that e.g. smooth functions should be more likely.
- We **would like** a certain type of functions.
- We have **prior knowledge** about specific properties, e.g. vanishing second derivative, etc.

Idea 2

We have to specify somehow, how likely it is to observe a specific function f from an overall class of functions. This is done by assuming some density $p(f)$ describing how likely we are to observe f.


2
1.5
1
0.5
0
−0.5
−1
−1.5
2
1
0
−1
−2
−2
−1
0
1
2


2
1.5
1
0.5
0
−0.5
−1
−1.5
2
1
0
−1
−2
−2
−1
0
1
2


2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
2
1
0
−1
−2
−2
−1
0
1
2

Examples

Speech Signal

We know that the signal is bandlimited, hence any signal containing frequency components above 10kHz has density 0.

Parametric Prior

We may know that f is a linear combination of $\sin x, \cos x, \sin 2x,$ and $\cos 2x$ and that the coefficients may be chosen from the interval $[-1, 1]$.

$$p(f) = \begin{cases} \frac{1}{16} & \text{if } f = \alpha_1 \sin x + \alpha_2 \cos x + \alpha_3 \sin 2x + \alpha_4 \cos 2x \text{ with } \alpha_i \in [-1, 1] \\ 0 & \text{otherwise} \end{cases}$$

Prior on Function Values

We assume that there is a correlation between the function values f_i at location $f(\mathbf{x}_i)$. There we have

$$p(f_1, f_2, f_3) = \frac{1}{\sqrt{(2\pi)^3 \det K}} \exp\left(-\frac{1}{2}(f_1, f_2, f_3)^\top K^{-1}(f_1, f_2, f_3)\right).$$

Prior on Function Values

Covariance Matrix

We may assume that the function values $f(\mathbf{x}_i)$ are correlated and follow a normal distribution with zero mean and covariance matrix K. In other words

$$f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m) \sim \mathcal{N}(0, K^{-1})$$

This means that for thes observations, we can write the prior distribution of $\mathbf{f} := f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m)$ as

$$p(\mathbf{f}) = (2\pi)^{-\frac{m}{2}} (\det K)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\, \mathbf{f}^\top K^{-1} \mathbf{f}\right).$$

Covariance Function

We posit that the matrix K is given by a covariance function k via $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Later we will see that k is identical to the kernel function of SVM.

How to use Priors

Bayes Rule

We want to infer the probability of f, having observed X, Y. By Bayes' rule we obtain

$$p(f|X,Y) = \frac{p(Y|f,X)p(f|X)}{p(Y|X)}.$$

This is also often called the **posterior probability** of observing f, after that the data X, Y arrived.

Usual Assumption

Typically we assume that X has no influence as to which f we may assume, i.e. $p(f|X) = p(f)$ (X and f are independent random variables).

Likelihood

$p(Y|f,X)$ is the Likelihood term that we used in Maximum Likelihood estimation. All that is happening is a **reweighting** of the likelihood by the prior distribution.

Inference

Goal

We want to infer f, possibly its value at a new location $\mathbf{x}$ via $p(f|X,Y)$.

Trick

The quantity $p(Y|X)$ is usually quite hard to obtain, moreover it is independent of f, therefore we can just treat it as a normalizing factor and we obtain

$$p(f|X,Y) \propto p(Y|f,X)p(f)$$

The normalization constant can be taken care of later.

Prediction

If we want to compute the expected value of $f(\mathbf{x})$ at a new location all we have to do is compute

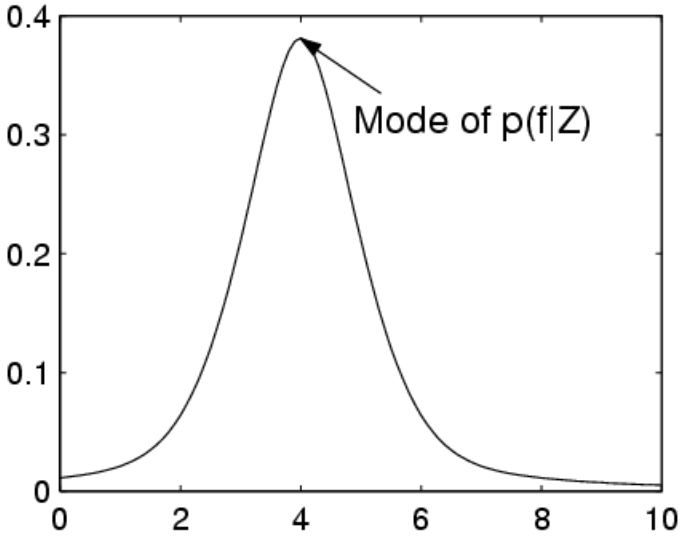
$$\mathbf{E}[f(\mathbf{x})] = \int f(\mathbf{x})p(f|X,Y)df$$

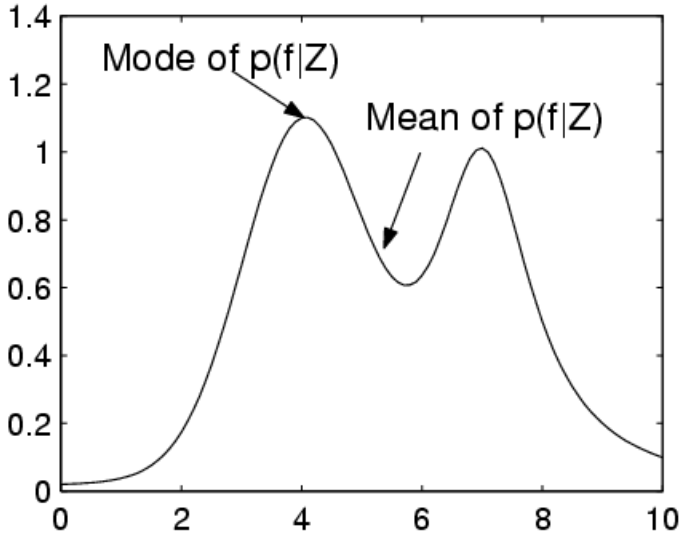
Variance

Likewise, to infer the predictive variance we compute

$$\mathbf{E}\left[(f(\mathbf{x}) - \mathbf{E}[f(\mathbf{x})])^2\right] = \int (f(\mathbf{x}) - \mathbf{E}[f(\mathbf{x})])^2 \, p(f|X,Y)df$$

This means that we can estimate the variation of $f(\mathbf{x})$, given the data and our prior knowledge about f, as encoded by $p(f)$.





Approximate Inference

Problem

Nobody wants to compute integrals ...

Idea

After all, we are only **averaging**, so replace the mean of the distribution by the mode and hope that it will be ok. This leads to the maximum a posteriori estimate (see next slide).

Lucky Coincidence

For Gaussian distributions (and many others) mode and mean coincide.

Problem 2

For some distributions it does not work well ...

Idea 2

Approximate the posterior $p(f|X,Y)$ by a **parametric** model. This is often referred to as **variational approximation**.

Maximum a Posteriori Estimate

Maximizing the Posterior Probability

To find the hypothesis f with the highest posterior probability we have to maximize

$$p(f|X,Y) = \frac{p(Y|f,X)p(f|X)}{p(Y|X)}$$

Lazy Trick

Since we only want f (and $p(Y|X)$ is independent of f), all we have to do is maximize $p(Y|f,X)p(f)$.

Taking Logs

For convenience we get f by minimizing

$$-\log p(Y|f,X)p(f|X) = -\log p(Y|f,X) - \log p(f) = -\log \mathcal{L} - \log p(f)$$

So all we are doing is to **reweight the likelihood** by $-\log p(f)$. This looks suspiciously like the regularization term. We will match up the two terms later.

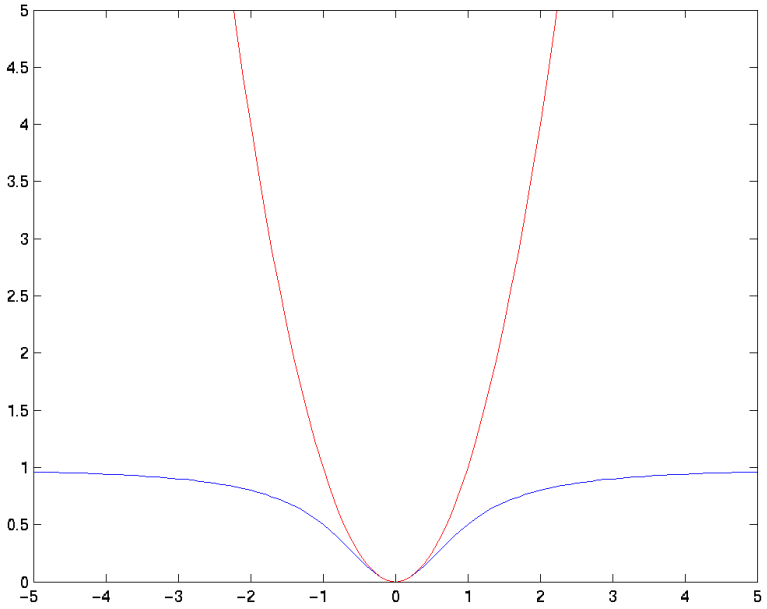
Maximum a Posteriori Estimate, Part II

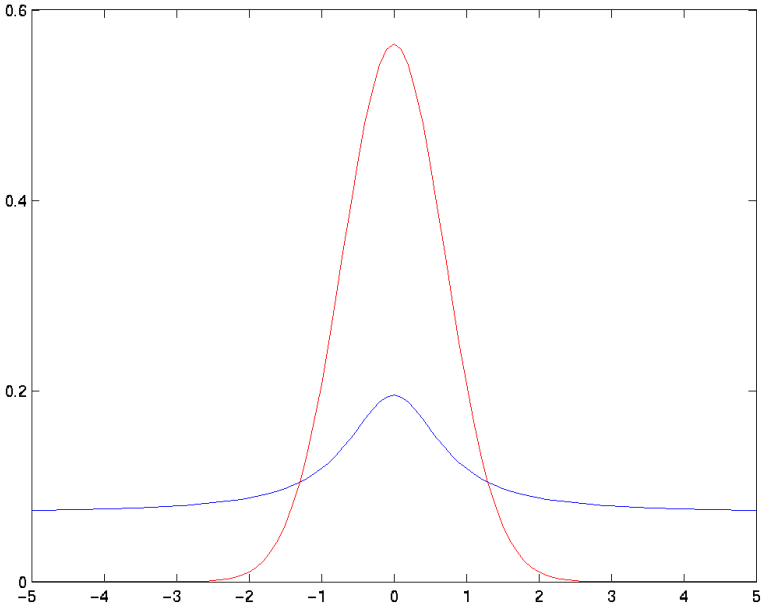
THE AUSTRALIAN NATIONAL UNIVERSITY

Variance

Once we found the **mode** f_0 of the distribution, we might as well approximate the variance by approximating $p(f|X,Y)$ with a normal distribution around f_0.

This is done by computing the second order information at f_0, i.e. $\partial_f^2 - \log p(f|X,Y)$.





Connection to Regularized Risk

Recycling of the Likelihood

Match up terms as we did with the likelihood and the loss function. In particular, we recycle these terms:

$$\begin{aligned} c(\mathbf{x}, y, f(\mathbf{x})) &\equiv -\log p(y - f(\mathbf{x})) \\ p(y|f(\mathbf{x}) &\equiv \exp(-c(\mathbf{x}, y, f(\mathbf{x})) \end{aligned}$$

Now all we have to do is take care of $m\lambda\Omega[f]$ and $-\log p(f)$.

Regularizer and Prior

The correspondence

$$m\lambda\Omega[f] + c = -\log p(f) \text{ or equivalently } p(f) \propto \exp(-m\lambda\Omega[f])$$

is the link between regularizer and prior.

Caveat

The translation from regularizer into prior works only to some extent, since the integral over f need not converge.

Posterior Distribution for Regression with Additive Gaussian Noise

$$p(\mathbf{f}|Z) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^{m}(y_i - f(\mathbf{x}_i))^2\right)\exp\left(-\frac{1}{2}\mathbf{f}^\top K^{-1}\mathbf{f}\right)$$

If we reparametrize $\mathbf{f} = K\alpha$ we have

$$p(\alpha|Z) \propto \exp\left(-\frac{1}{2}\|\mathbf{y} - K\alpha\|^2 - \frac{1}{2}\alpha^\top K\alpha\right)$$

Regularized Risk Functional for LMS Regression

$$R_{\text{reg}}[f] = \sum_{i=1}^{m}(y_i - f(\mathbf{x}_i))^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

If we reparametrize $\mathbf{w} = \sum_{i=1}^{m}\alpha_i\Phi(\mathbf{x}_i)$ we have

$$R_{\text{reg}}[f] = -\frac{1}{2}\|\mathbf{y} - K\alpha\|^2 + \frac{\lambda}{2}\alpha^\top K\alpha$$

Taking negative logs, we can see that GP and SVM lead to the same maximum a posteriori solution.

Summary

- Learning as risk minimization
- Feature Spaces and Kernel Trick
- The ν-Trick
- Chunking and SMO
- Online Learning
- Bayesian Methods

For more information see

http://www.kernel-machines.org

or ... shameless plug ... buy the book

Schölkopf and Smola: Learning with Kernels

MIT Press, November 2001