SPARSE KERNEL MACHINES

J. Elder

Motivation

Inference can be slow for kernel methods, as the kernel $k(\mathbf{x},\mathbf{x}_n)$ must be evaluated for the new data point $\mathbf{x}$ against **all** training data points $\mathbf{x}_n$.

In a sparse kernel machine, the kernel $k(\mathbf{x},\mathbf{x}_n)$ need only be evaluated for a subset of the training data.

We will focus in particular on the **Support Vector Machine** (SVM), applied to **classification** problems.

SVMs are **discriminative decision machines**:
they do not provide posterior probabilities.

Support Vector Machines

SVMs are based on the linear model $y(\mathbf{x}) = \mathbf{w}^t \phi(\mathbf{x}) + b$

Assume training data $\mathbf{x}_1, \ldots, \mathbf{x}_N$ with coresponding target values $t_1, \ldots, t_N$, $\mathrm{t}_n \in \{-1, 1\}$.

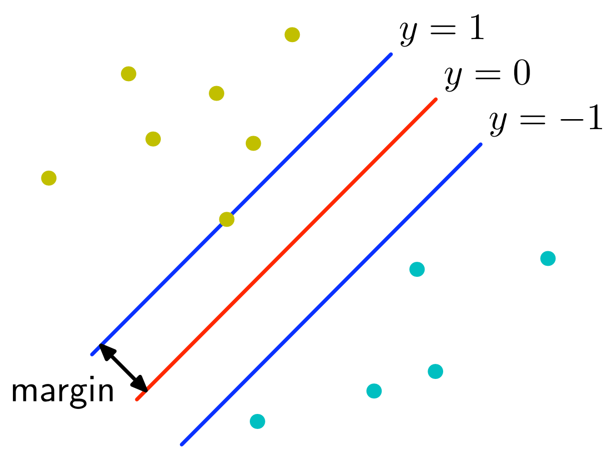
$\mathbf{x}$ classified according to sign of $y(\mathbf{x})$.

Assume for the moment that the training data are linearly separable in feature space.

Then $\exists \mathbf{w}, b : t_n y\left(\mathbf{x}_n\right) > 0 \ \forall n \in [1, \ldots N]$

Maximum Margin Classifiers

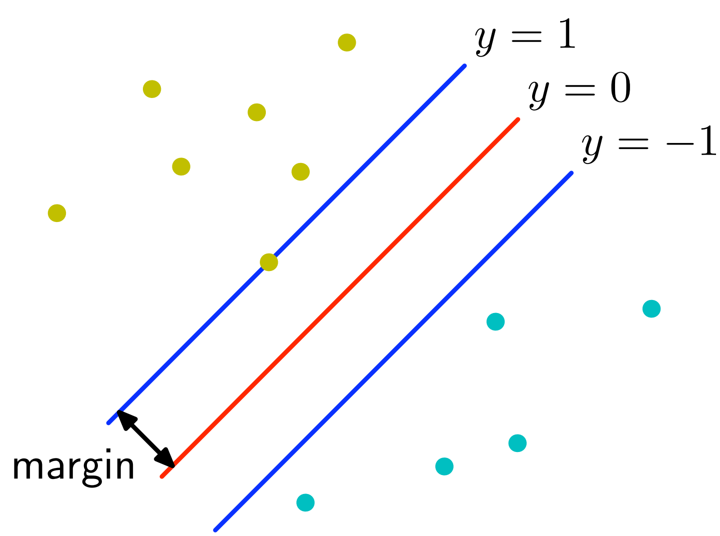
- When the training data are linearly separable, there are generally an infinite number of solutions for ($\mathbf{w}$, b) that separate the classes exactly.
- The **margin** of such a classifier is defined as the orthogonal distance in feature space between the decision boundary and the closest training vector.
- SVMs are an example of a **maximum margin classifer**, which finds the linear classifier that maximizes the margin.



Probabilistic Motivation

- The maximum margin classifier has a probabilistic motivation.

If we model the class-conditional densities with a KDE using Gaussian kernels with variance σ^2, then in the limit as $\sigma \to 0$, the optimal linear decision boundary $\to$ maximum margin linear classifier.



Two Class Discriminant Function

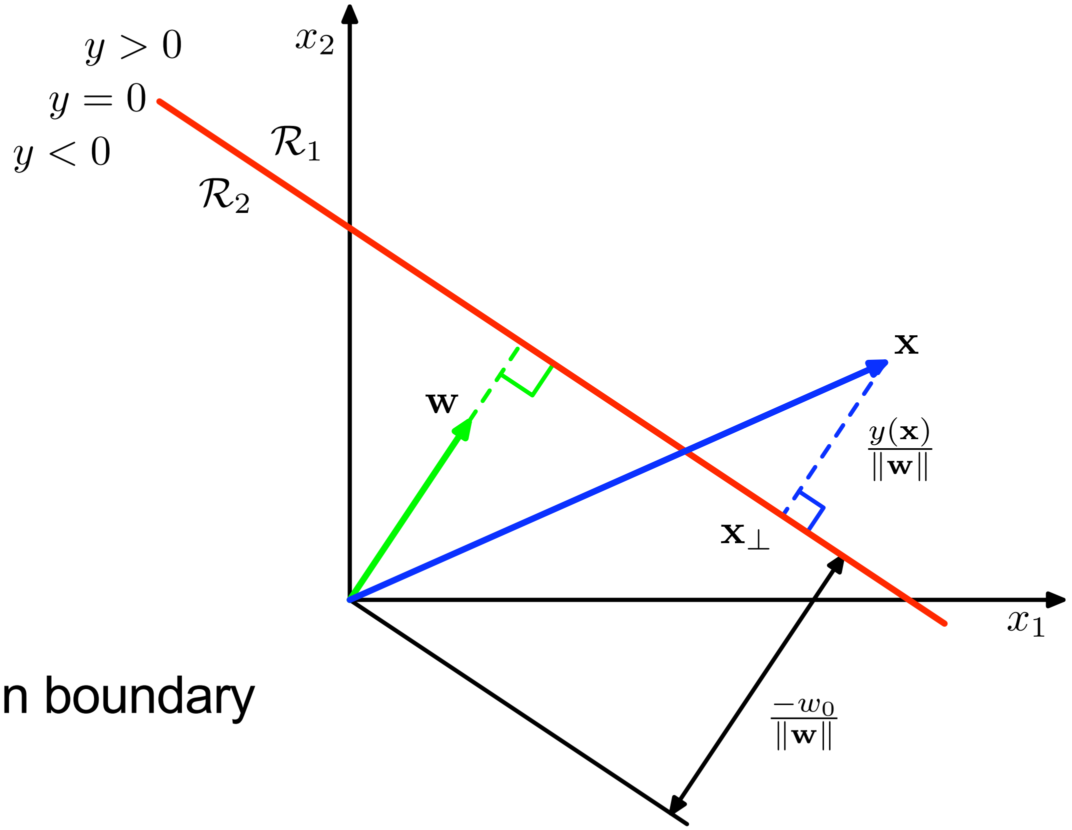
Let $f(\cdot)$ be the identity:

$$y(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0$$

$$y(\mathbf{x}) \geq 0 \rightarrow \mathbf{x} \text{ assigned to } C_1$$

$$y(\mathbf{x}) < 0 \rightarrow \mathbf{x} \text{ assigned to } C_2$$

Thus $y(\mathbf{x}) = 0$ defines the decision boundary



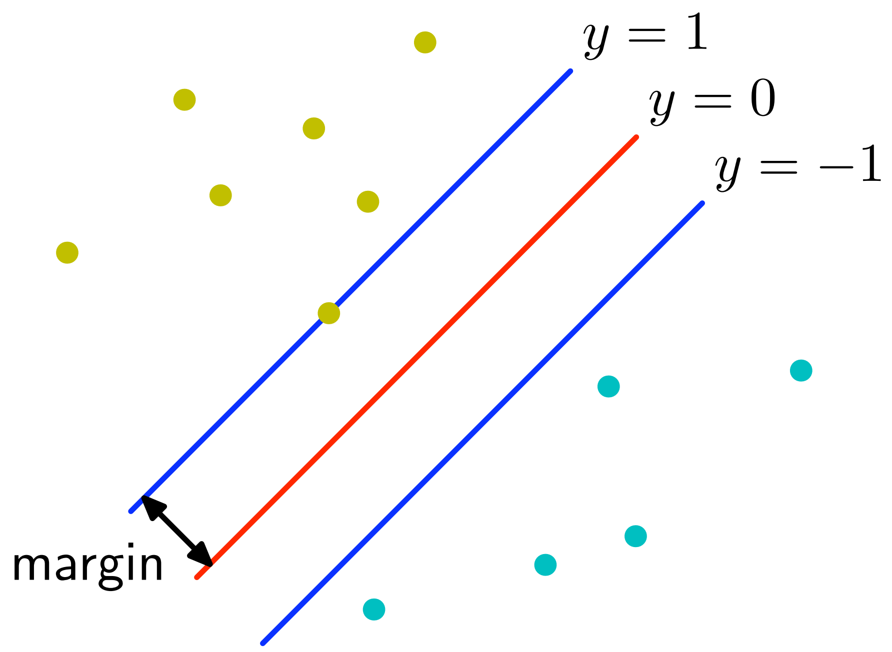
Maximum Margin Classifiers

Distance of point $\mathbf{x}_n$ from decision surface is given by:

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n\left(\mathbf{w}^t\phi(\mathbf{x}_n)+b\right)}{\|\mathbf{w}\|}$$

Thus we seek:

$$\underset{\mathbf{w},b}{\arg\max}\left\{\frac{1}{\|\mathbf{w}\|}\min_n\left[t_n\left(\mathbf{w}^t\phi(\mathbf{x}_n)+b\right)\right]\right\}$$



Maximum Margin Classifiers

Distance of point $\mathbf{x}_n$ from decision surface is given by:

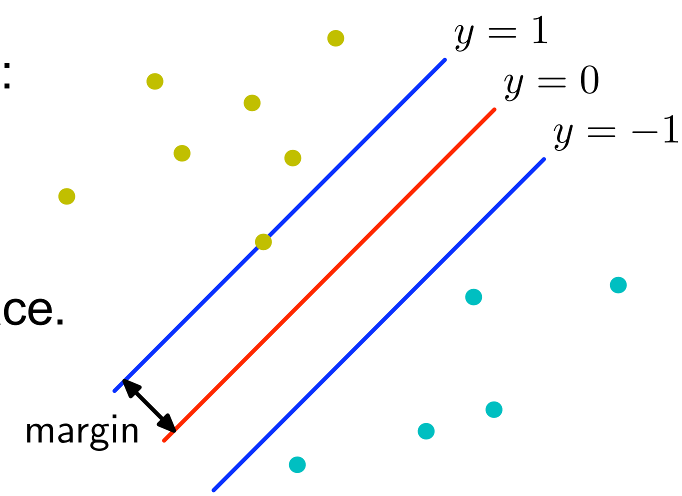
$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n\left(\mathbf{w}^t \phi(\mathbf{x}_n) + b\right)}{\|\mathbf{w}\|}$$

Note that rescaling $\mathbf{w}$ and b by the same factor leaves the distance to the decision surface unchanged.

Thus, wlog, we consider only solutions that satisfy:

$$t_n\left(\mathbf{w}^t \phi(\mathbf{x}_n) + b\right) = 1.$$

for the point $\mathbf{x}_n$ that is closest to the decision surface.



Quadratic Programming Problem

Then all points $\mathbf{x}_n$ satisfy $t_n\left(\mathbf{w}^t\phi(\mathbf{x}_n)+b\right)\geq 1$

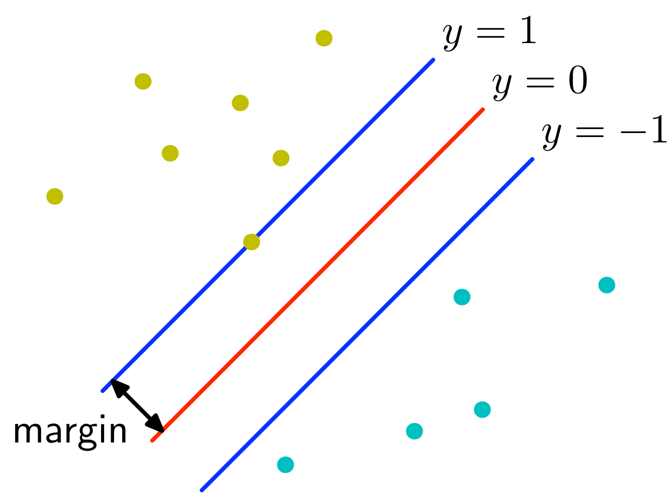
Points for which equality holds are said to be **active**.
All other points are **inactive**.

Now $\underset{\mathbf{w},b}{\arg\max}\left\{\frac{1}{\|\mathbf{w}\|}\min_n\left[t_n\left(\mathbf{w}^t\phi(\mathbf{x}_n)+b\right)\right]\right\}$

$\leftrightarrow \frac{1}{2}\underset{\mathbf{w}}{\arg\min}\|\mathbf{w}\|^2$

Subject to $t_n\left(\mathbf{w}^t\phi(\mathbf{x}_n)+b\right)\geq 1\ \forall \mathbf{x}_n$

This is a **quadratic programming** problem.

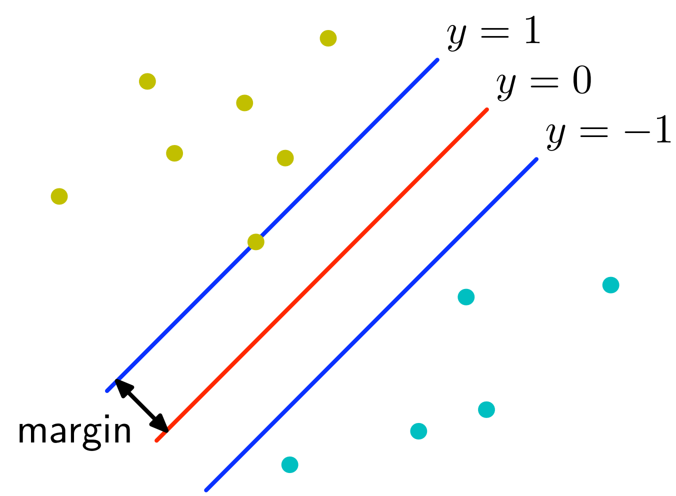


Quadratic Programming Problem

$$\frac{1}{2}\underset{\mathbf{w}}{\arg\min}\|\mathbf{w}\|^2, \text{ subject to } t_n\left(\mathbf{w}^t\phi\left(\mathbf{x}_n\right)+b\right)\geq 1\ \forall \mathbf{x}_n$$

Solve using Lagrange multipliers a_n :

$$L(\mathbf{w},b,\mathbf{a})=\frac{1}{2}\underset{\mathbf{w}}{\arg\min}\|\mathbf{w}\|^2-\sum_{n=1}^{N}a_n\left\{t_n\left(\mathbf{w}^t\phi\left(\mathbf{x}_n\right)+b\right)-1\right\}$$



END OF LECTURE NOV 8, 2010

Dual Representation

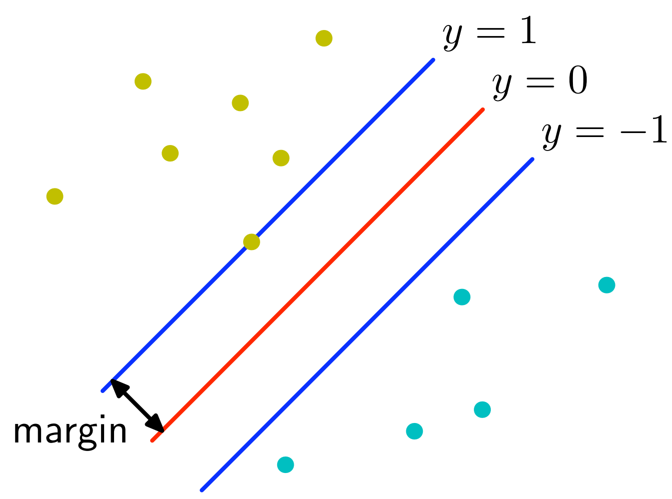
Solve using Lagrange multipliers a_n :

$$L(\mathbf{w},b,\mathbf{a}) = \frac{1}{2}\underset{\mathbf{w}}{\arg\min}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left\{ t_n \left(\mathbf{w}^t \phi\left(\mathbf{x}_n\right) + b \right) - 1 \right\}$$

Setting derivatives with respect to **w** and b, we get:

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n)$$

$$\sum_{n=1}^{N} a_n t_n = 0$$



Dual Representation

Substituting for **w** and b leads to the dual representation of the maximum margin problem, in which we maximize:

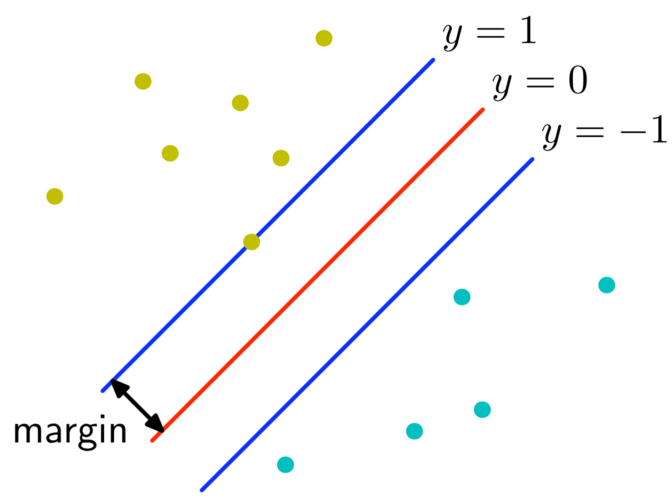
$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

with respect to **a**, subject to:

$$a_n \geq 0 \ \forall n$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

and where $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^t \phi(\mathbf{x}')$



Dual Representation

Using $\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n)$, a new point x is classified by computing

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b$$

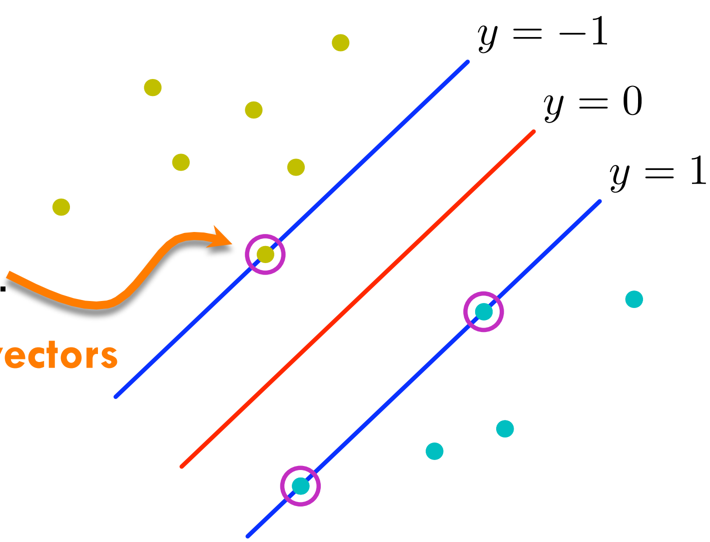
The Karush-Kuhn-Tucker (KKT) conditions for this constrained optimization problem are:

$$a_n \geq 0$$

$$t_n y(\mathbf{x}_n) - 1 \geq 0$$

$$a_n \left\{ t_n y(\mathbf{x}_n) - 1 \right\} = 0$$

Thus for every data point, either $a_n = 0$ or $t_n y(\mathbf{x}_n) = 1$.



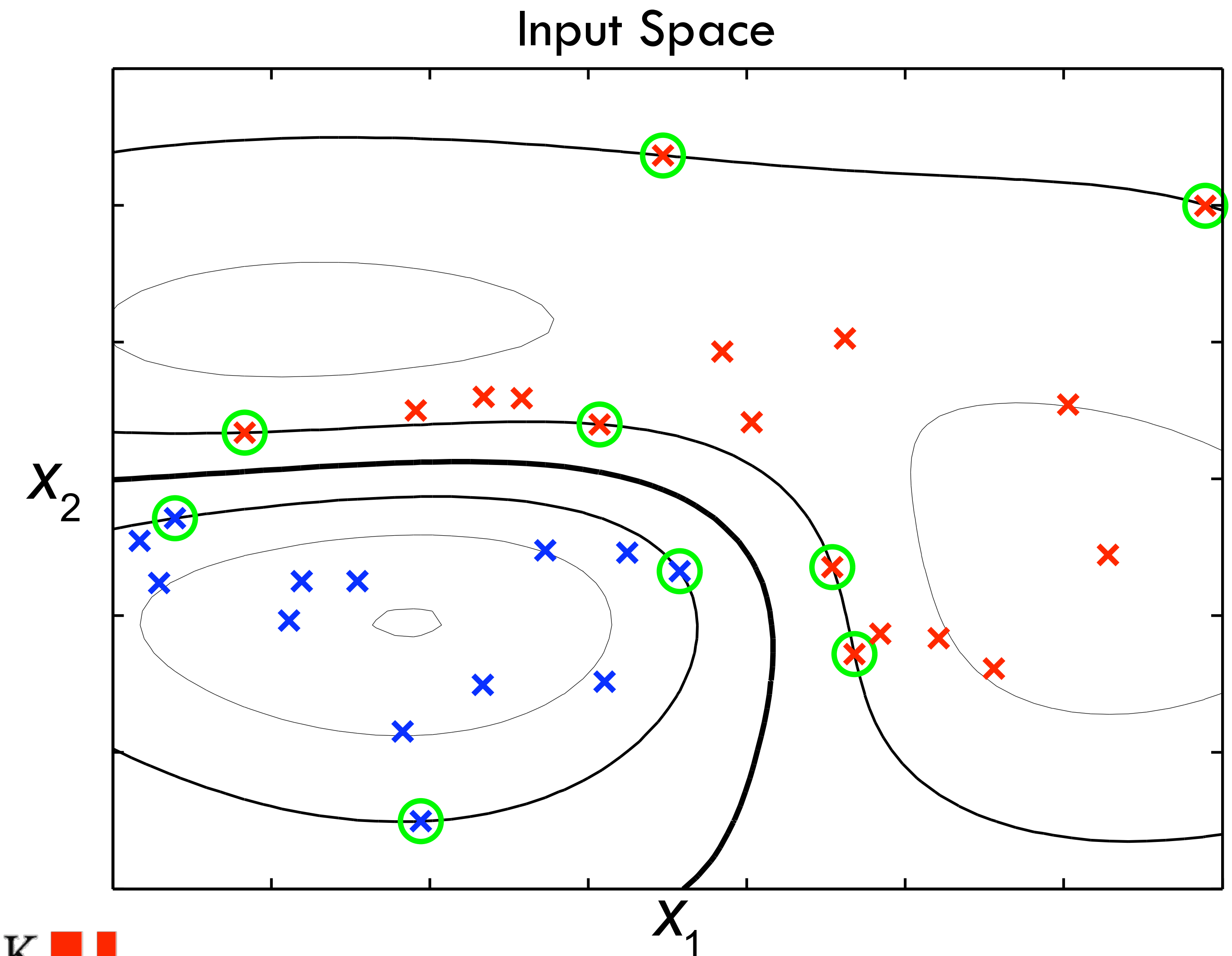
Solving for the Bias

Once the optimal **a** is determined, the bias *b* can be computed from

$$b = \frac{1}{N_S} \sum_{n \in S} \left(t_n - \sum_{m \in S} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right)$$

where *S* is the index set of support vectors and N_S is the number of support vectors.

Example (Gaussian Kernel)



Overlapping Class Distributions

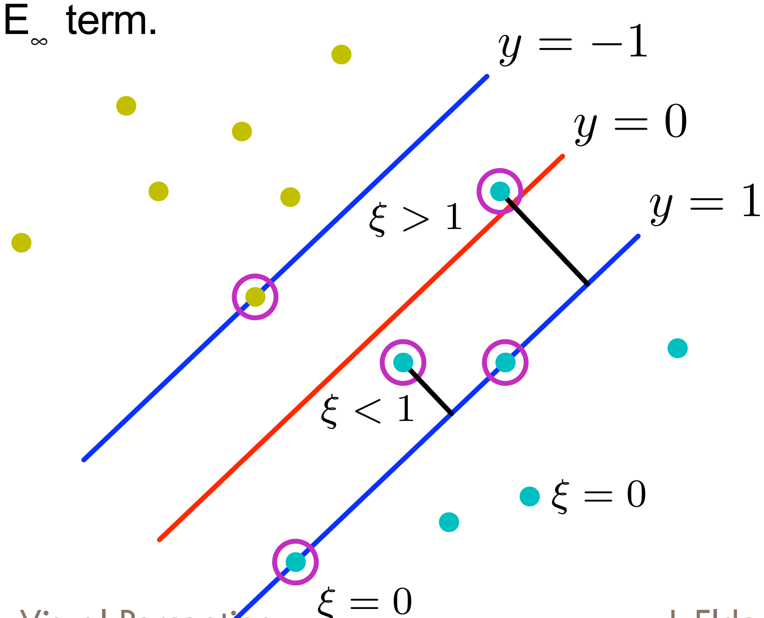
The SVM for non-overlapping class distributions can be expressed as the minimization of

$$\sum_{n=1}^{N} E_\infty\left(y\left(\mathbf{x}_n\right)t_n - 1\right) + \lambda\left\|\mathbf{w}\right\|^2$$

where $E_\infty(z)$ is 0 if $z \geq 0$, and ∞ otherwise.

This forces all points to lie on or outside the margins, on the correct side for their class.

To allow for misclassified points, we have to relax this E_∞ term.



Slack Variables

To this end, we introduce N **slack variables** $\xi_n \geq 0,\; n = 1,\ldots N$.

$\xi_n = 0$ for points on or on the correct side of the margin boundary for their class

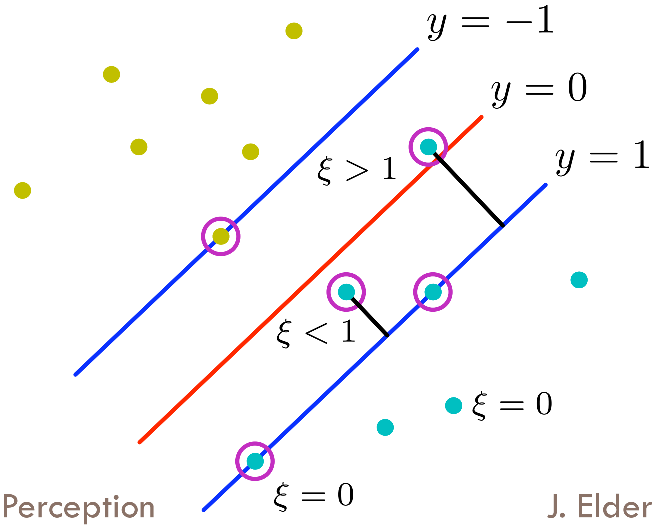
$\xi_n = \left| t_n - y\left(\mathbf{x}_n\right) \right|$ for all other points.

Thus $\xi_n < 1$ for points that are correctly classified

$\xi_n > 1$ for points that are incorrectly classified

We now minimize $C\sum_{n=1}^{N} \xi_n + \frac{1}{2}\left\|\mathbf{w}\right\|^2$, where $C > 0$.

subject to $t_n y\left(\mathbf{x}_n\right) \geq 1 - \xi_n$, and $\xi_n \geq 0,\; n = 1,\ldots N$



Dual Representation

This leads to a dual representation, where we maximize

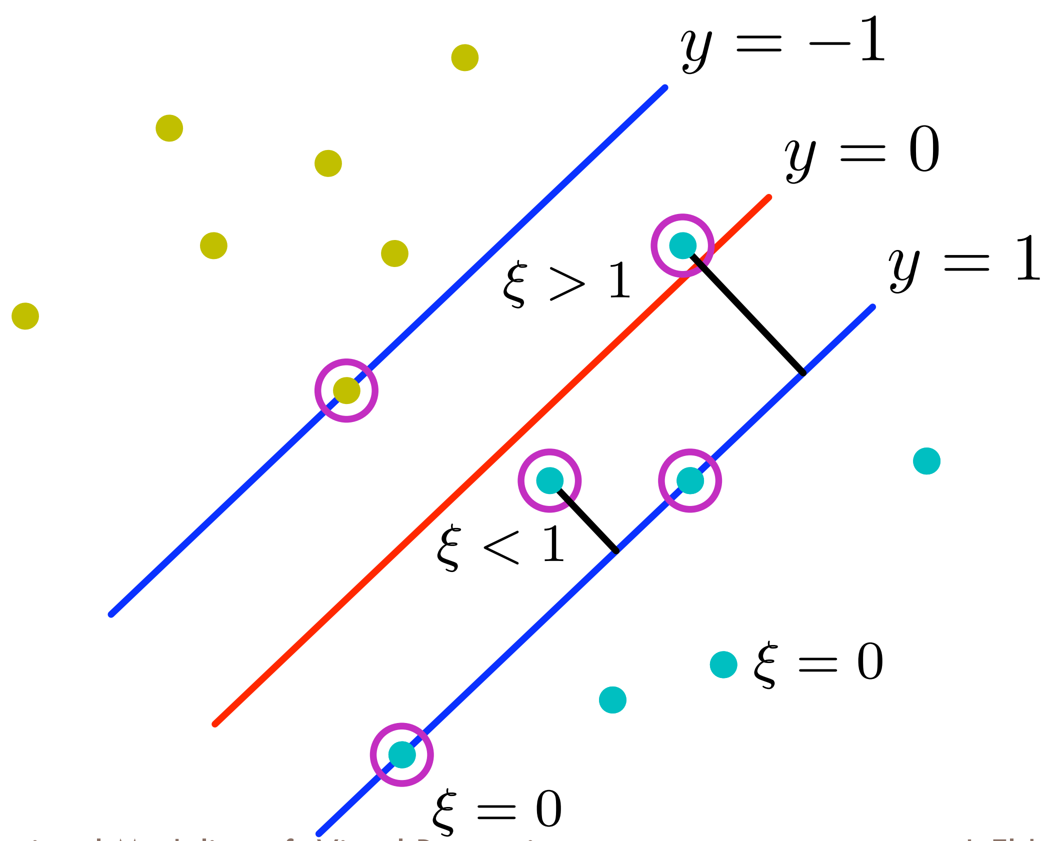
$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k\left(\mathbf{x}_n, \mathbf{x}_n\right)$$

with constraints

$$0 \le a_n \le C$$

and

$$\sum_{n=1}^{N} a_n t_n = 0$$



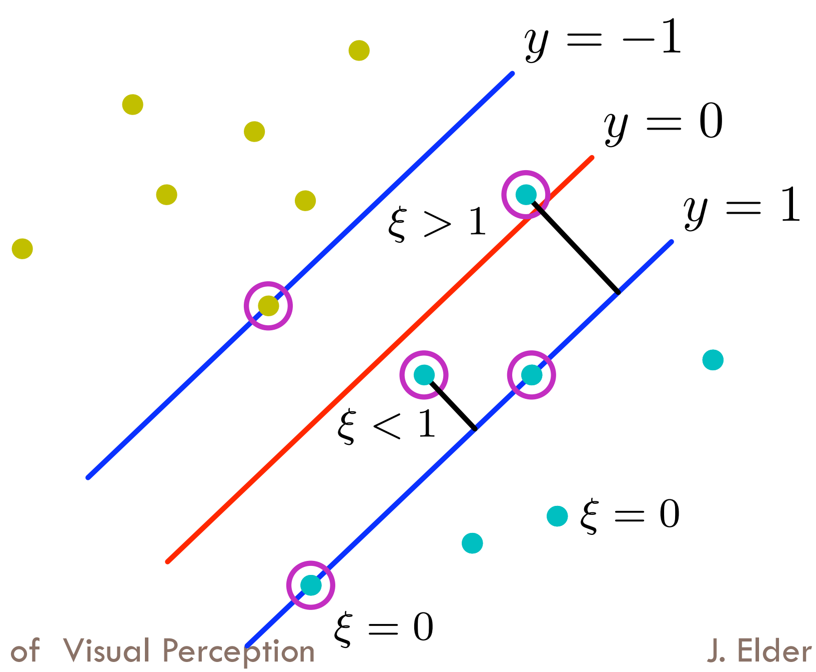
Support Vectors

Again, a new point **x** is classified by computing

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b$$

For points that are on the correct side of the margin, $a_n = 0$.

Thus support vectors consist of points between their margin and the decision boundary, as well as misclassified points.



Bias

Again, a new point **x** is classified by computing

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b$$

Once the optimal **a** is determined, the bias *b* can be computed from

$$b = \frac{1}{N_{\mathcal{M}}} \sum_{n \in \mathcal{M}} \left(t_n - \sum_{m \in S} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right)$$

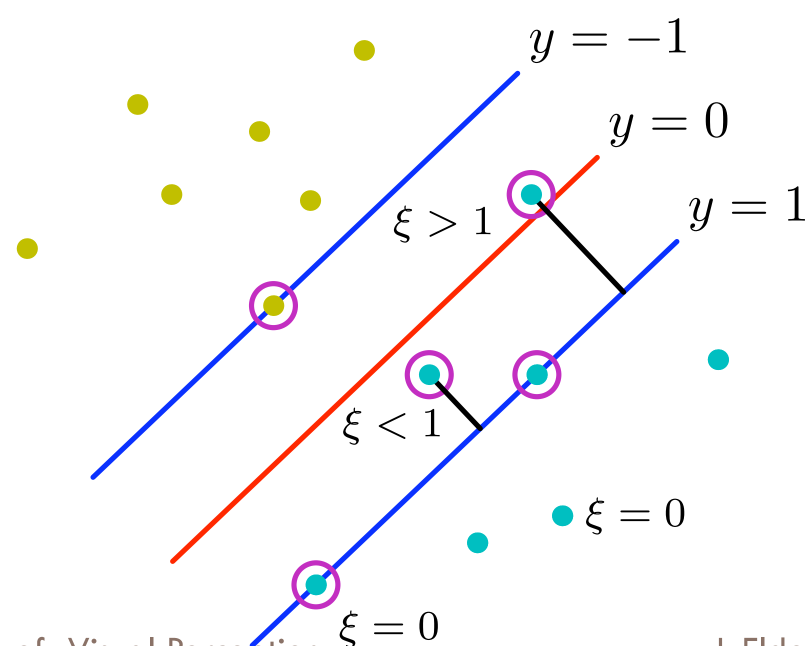
where

S is the index set of support vectors

N_S is the number of support vectors

$\mathcal{M}$ is the index set of points on the margins

$N_{\mathcal{M}}$ is the number of points on the margins



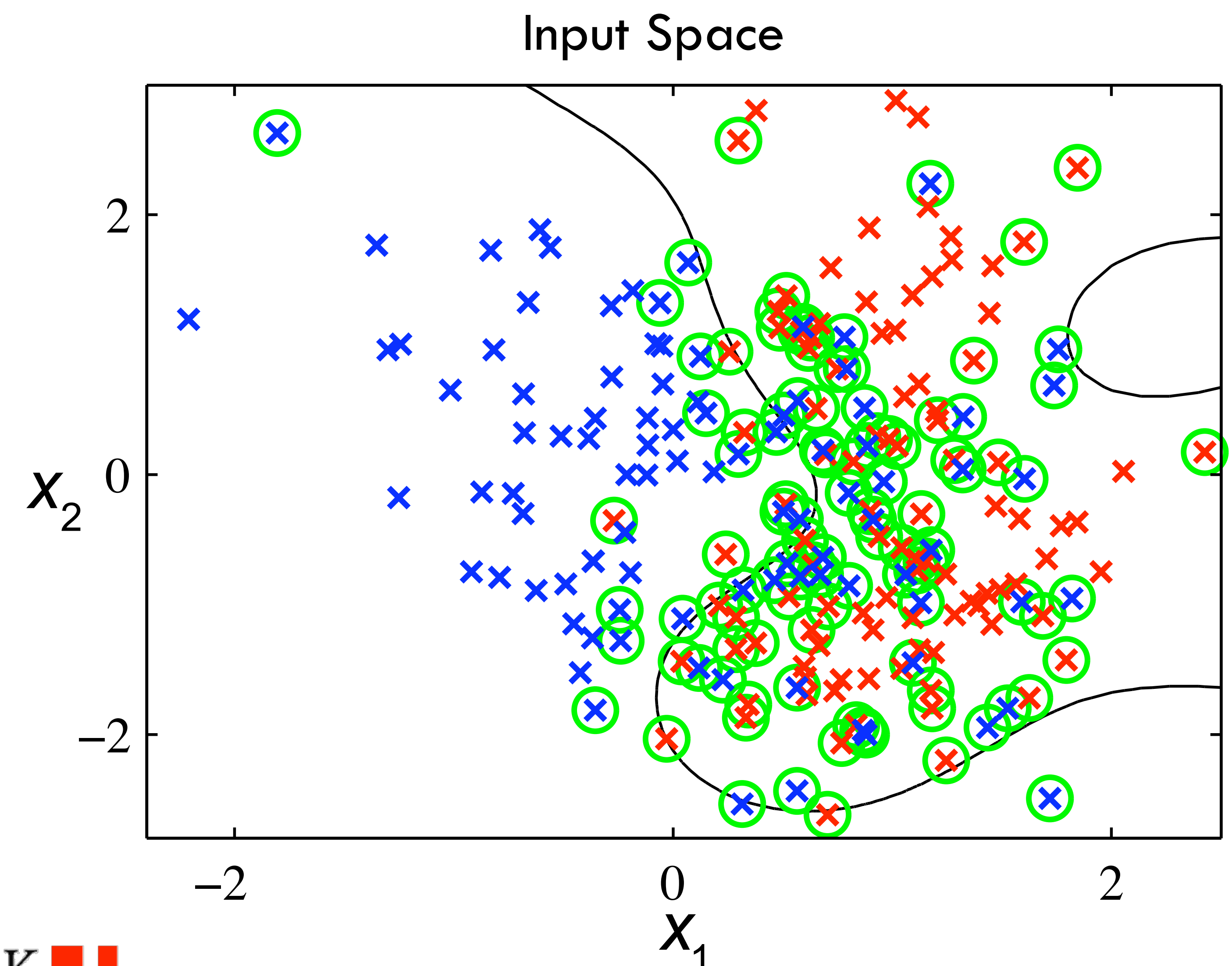
Solving the Quadratic Programming Problem

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k\left(\mathbf{x}_n, \mathbf{x}_n\right)$$

subject to $0 \le a_n \le C$ and $\sum_{n=1}^{N} a_n t_n = 0$

- Problem is convex.
- Solutions are generally $O(N^3)$.
- Traditional quadratic programming techniques often infeasible due to computation and memory requirements.
- Instead, heuristic methods such as sequential minimal optimization can be used, that in practice are found to scale as $O(N)$ - $O(N^2)$.

Example



Relation to Logistic Regression

The objective function for the soft-margin SVM can be written as:

$$\sum_{n=1}^{N} E_{SV}\left(y_n t_n\right) + \lambda \|\mathbf{w}\|^2$$

where $E_{SV}(z) = [1 - z]_+$ is the **hinge error function,**

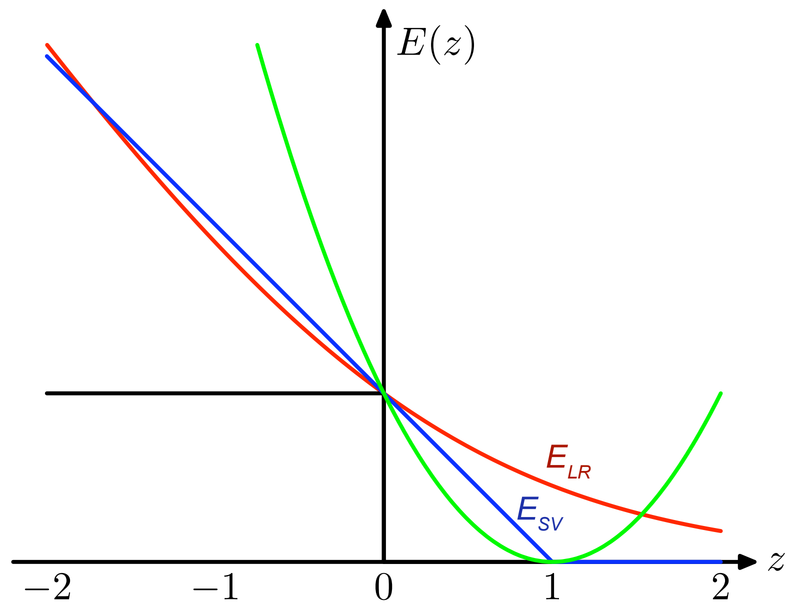
and $[z]_+ = z \textit{ if } z \geq 0$

$= 0$ otherwise.

For $t \in \{-1,1\}$, the objective function for a regularized version of logistic regression can be written as:

$$\sum_{n=1}^{N} E_{LR}\left(y_n t_n\right) + \lambda \|\mathbf{w}\|^2$$

where $E_{LR}(z) = \log\left(1 + \exp(-z)\right)$.

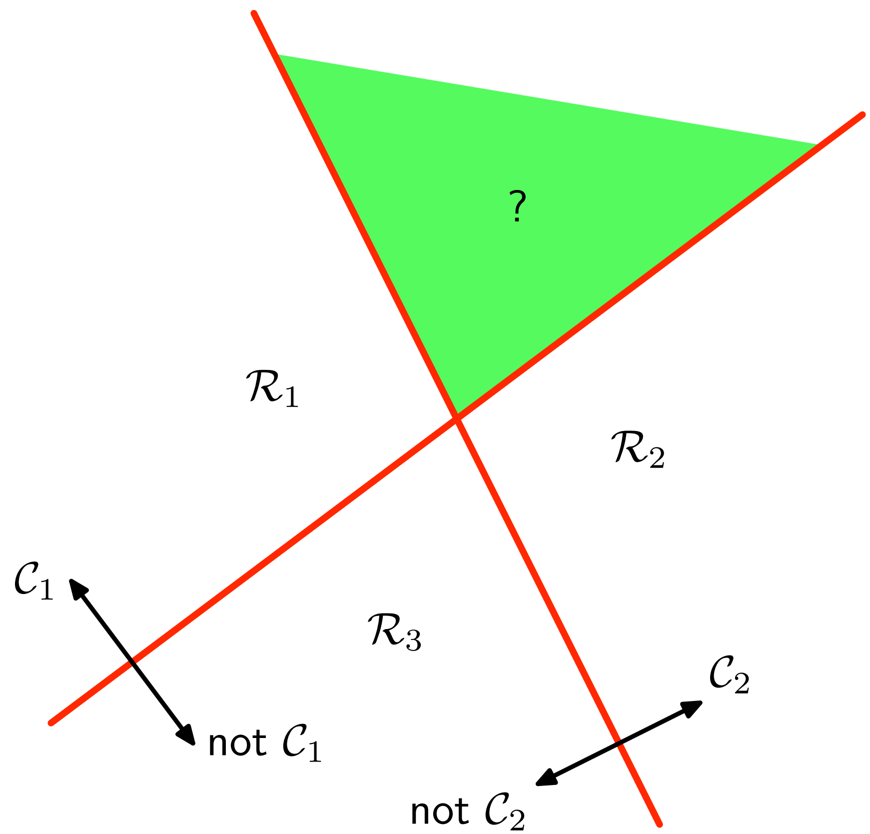


Multiclass SVMs

- We encounter the same problems we experienced with least-squares.

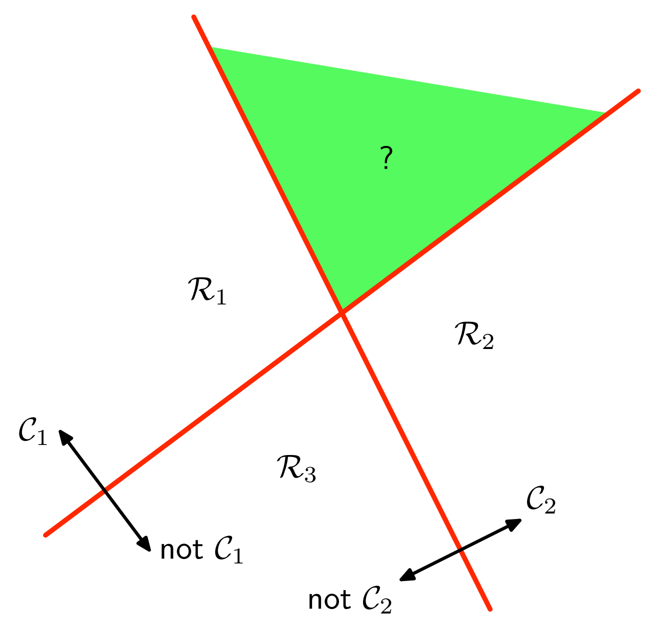
One-Versus-The-Rest

- Idea #1: Just use *K-1* discriminant functions, each of which separates one class C_k from the rest.
- Problem: Ambiguous regions



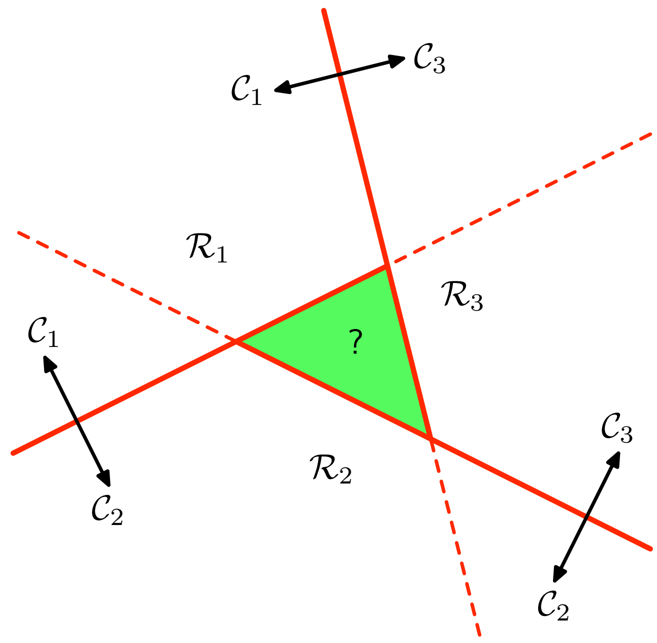
One-Versus-The-Rest

- Possible Solution: select class according to: $\arg\max_k y_k(\mathbf{x})$
- Problems:
  - Classifiers were all trained separately.
    - Methods for joint training have been proposed – slows training.
  - Training is imbalanced (e.g., for K=10 classes, 10% in-class, 90% out-of-class)
    - Can be solved by using $t_n \in \left\{ -\frac{1}{K-1}, 1 \right\}$.

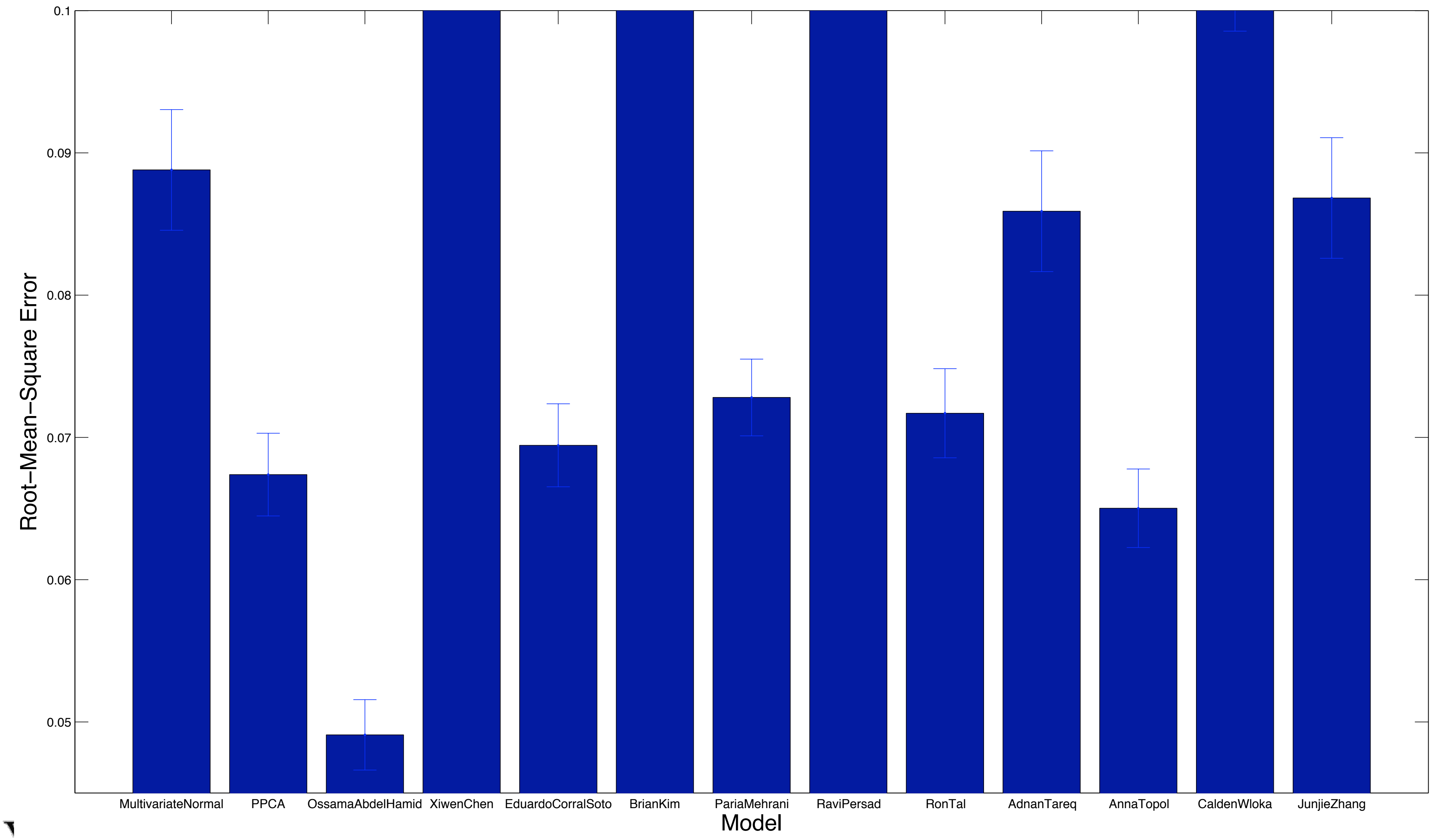


One-Versus-One

- Idea #2: Use K(*K*-*1*)/2 discriminant functions, each of which separates two classes C_j, C_k from each other.
- Each point classified by majority vote.
- Problems:
  - Ambiguous regions
  - Expensive



Assignment 1 Results



Methods Submitted

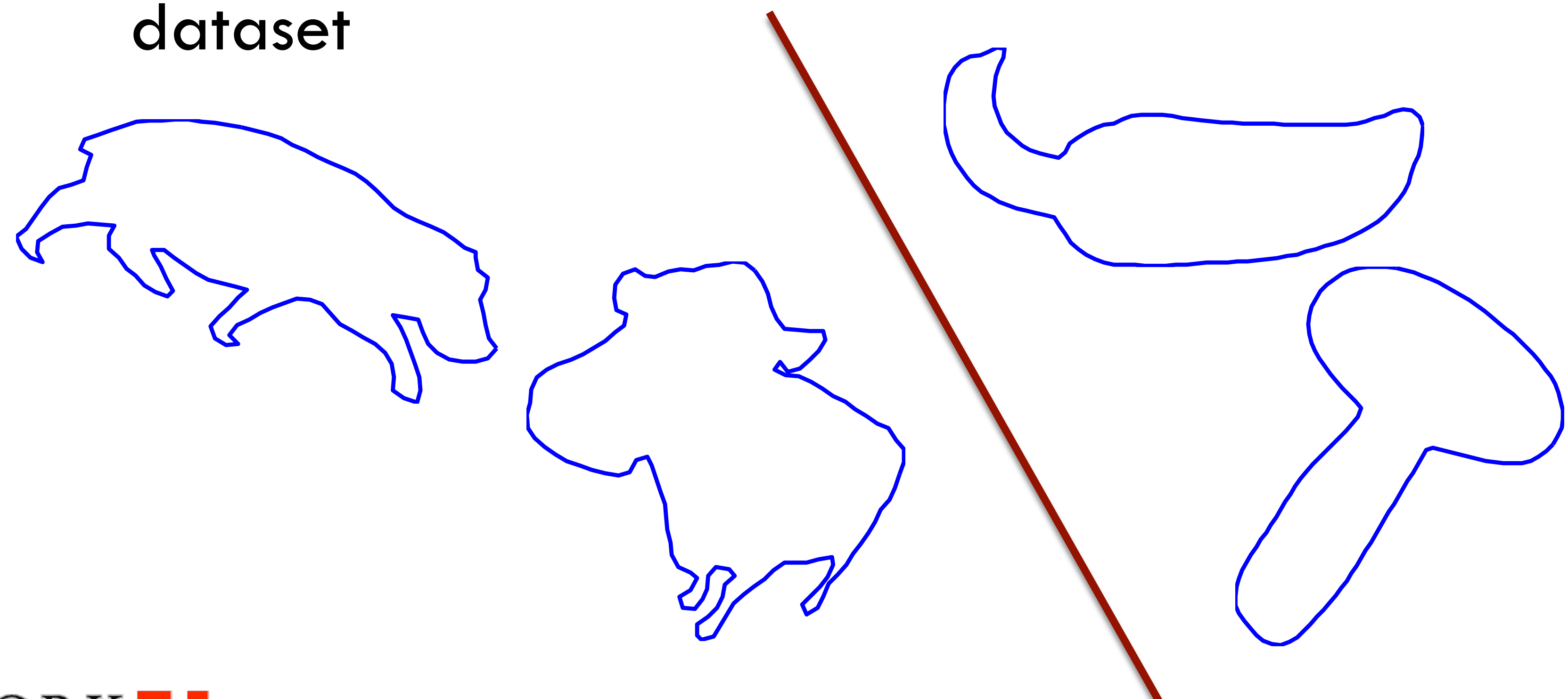
- Hierarchy of Gaussian models
- Treat x and y coordinates as independent
- Probabilistic PCA
- Gaussian mixtures
- Mean shift
- Use sample mean rather than theoretical mean
- Approximate mean as an ellipse
- Local Gaussian model
- Bi-arc interpolation

Some Things We've Learned

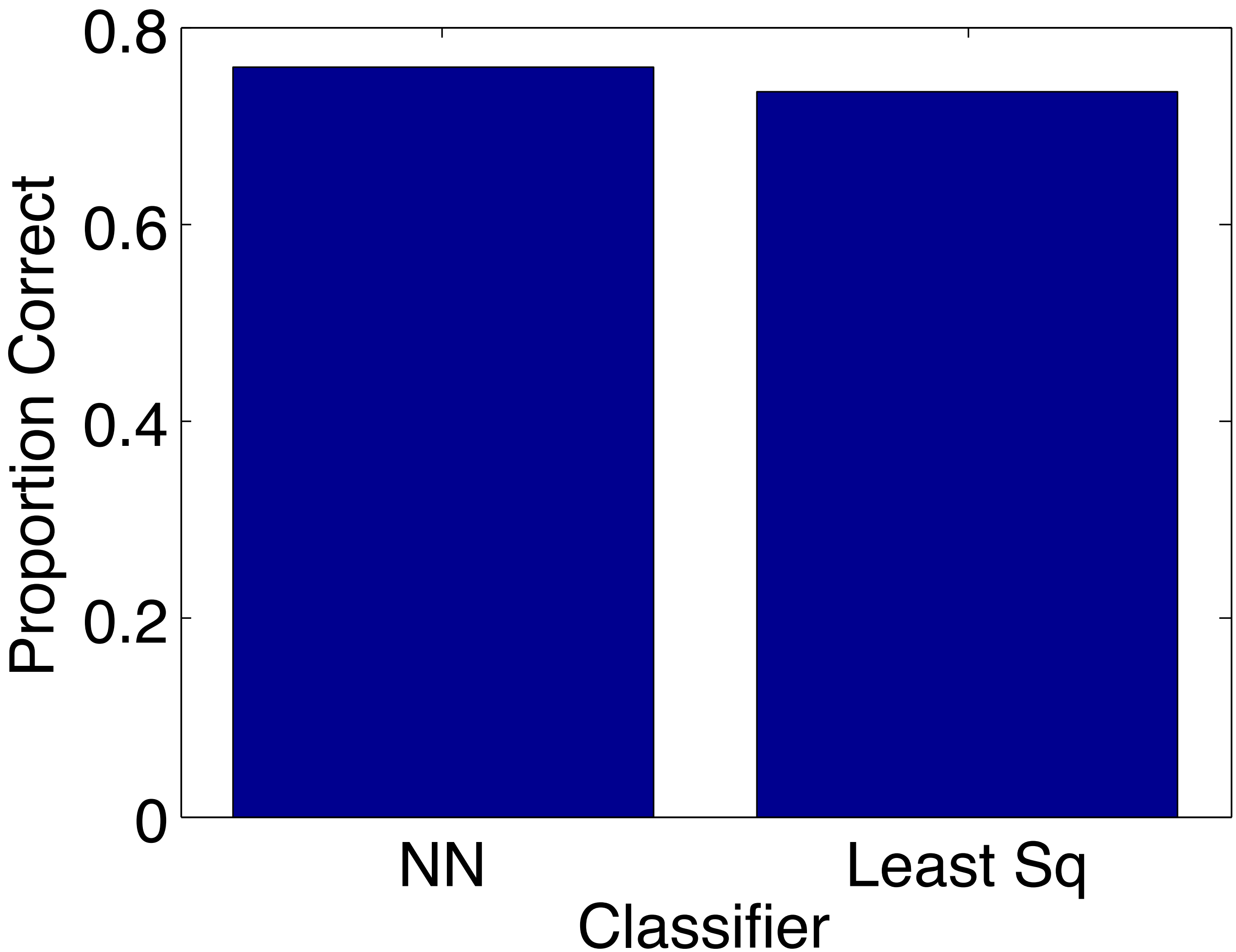
- Use the book!
- The curse of dimensionality
- Probabilistic PCA
- The importance of coding correctly!

Assignment 2

- Classify shapes as 'animal' or 'vegetable'
- Winner has the highest proportion correct
- May be tough to beat nearest-neighbour for this dataset



Classifiers Provided



SVMs for Regression

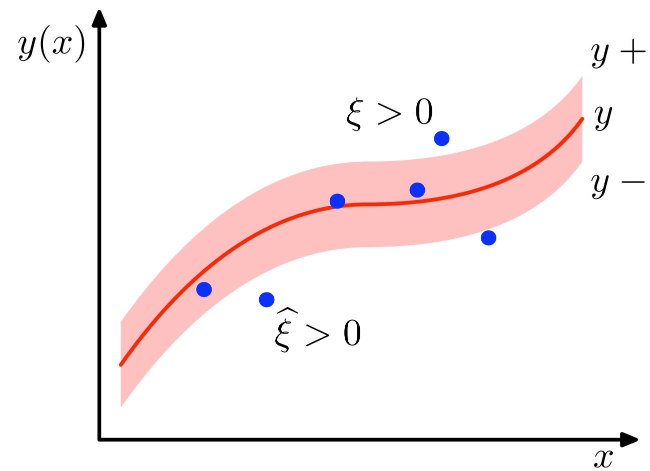
In standard linear regression, we minimize

$$\frac{1}{2}\sum_{n=1}^{N}\left(y_n - t_n\right)^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

This penalizes all deviations from the model.

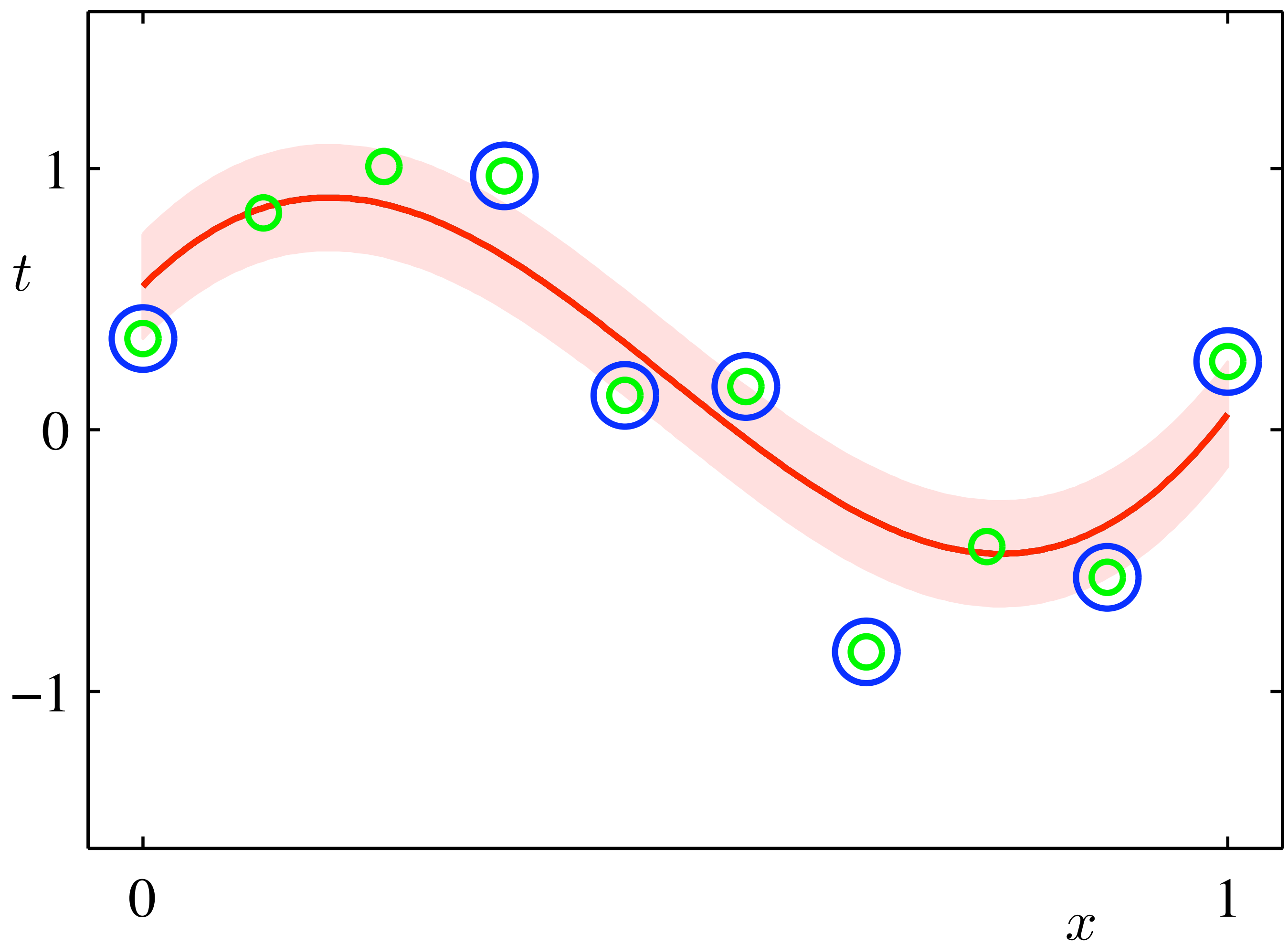
To obtain sparse solutions, we replace the quadratic error function by an ε-insensitive error function, e.g.,

$$E_\varepsilon\left(y(\mathbf{x}) - t\right) = \begin{cases} 0, \text{ if } \left|y(\mathbf{x}) - t\right| < \varepsilon \\ \left|y(\mathbf{x}) - t\right| - \varepsilon, \text{ otherwise} \end{cases}$$



See text for details of solution.

Example



Relevance Vector Machines

- Some drawbacks of SVMs:
  - Do not provide posterior probabilities.
  - Not easily generalized to $K > 2$ classes.
  - Parameters (C, ε) must be learned by cross-validation.
- The **Relevance Vector Machine** is a sparse Bayesian kernel technique that avoids these drawbacks.
- RVMs also typically lead to sparser models.

RVMs for Regression

$$p(t \mid \mathbf{x}, \mathbf{w}, \beta) = N\left(t \mid y(\mathbf{x}), \beta^{-1}\right)$$

where $y(\mathbf{x}) = \mathbf{w}^t \phi(\mathbf{x})$

In an RVM, the basis functions $\phi(\mathbf{x})$ are kernels $k\left(\mathbf{x}, \mathbf{x}_n\right)$:

$$y(x) = \sum_{n=1}^{N} w_n k\left(\mathbf{x}, \mathbf{x}_n\right) + b$$

However, unlike in SVMs, the kernels need not be positive definite, and the $\mathbf{x}_n$ need not be the training data points.

RVMs for Regression

Likelihood:

$$p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} p(t_n \mid \mathbf{x}_n, \mathbf{w}, \beta)$$

where the n^{th} row of $\mathbf{X}$ is $\mathbf{x}_n^t$.

Prior:

$$p(\mathbf{w} \mid \alpha) = \prod_{i=1}^{M} \mathcal{N}\left(w_i \mid 0, \alpha_i^{-1}\right)$$

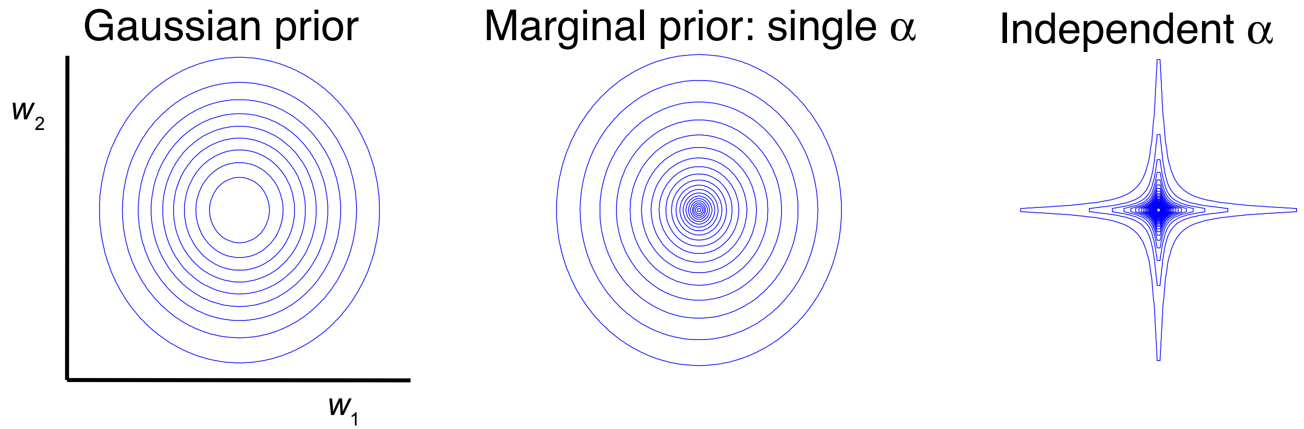
- Note that each weight parameter has its own precision hyperparameter.

RVMs for Regression

$$p(w_i \mid \alpha_i) = \mathcal{N}\left(w_i \mid 0, \alpha_i^{-1}\right)$$

$$p\left(\alpha_i\right) = \mathrm{Gam}\left(\alpha_i \mid a, b\right)$$

$$p\left(w_i\right) = \mathrm{St}\left(w_i \mid 0, a/b, 2a\right)$$



- The conjugate prior for the precision of a Gaussian is a gamma distribution.
- Integrating out the precision parameter thus leads to a Student's *t* distribution over w_i.
- Thus the distribution over **w** is a product of Student's t distributions.
- As a result, maximizing the evidence will yield a sparse **w**.
- Note that to achieve sparsity it is critical that each parameter w_i has a separate precision α_i.

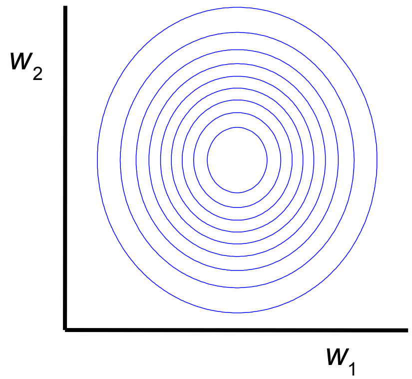
RVMs for Regression

$$p(w_i \mid \alpha_i) = N\left(w_i \mid 0, \alpha_i^{-1}\right)$$

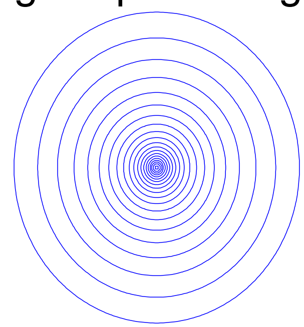
$$p\left(\alpha_i\right) = \mathrm{Gam}\left(\alpha_i \mid a, b\right)$$

$$p\left(w_i\right) = \mathrm{St}\left(w_i \mid 0, a/b, 2a\right)$$

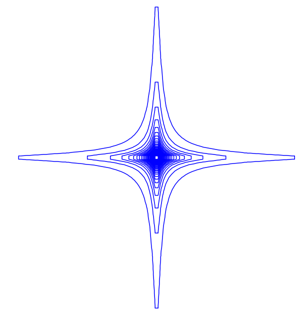
Gaussian prior



Marginal prior: single α



Independent α



If we let a $\to$ 0,b $\to$ 0, then $p\left(\log \alpha_i\right) \to$ uniform and $p\left(w_i\right) \to \left|w_i\right|^{-1}$.

RVMs for Regression

Likelihood:

$$p\left(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta\right) = \prod_{n=1}^{N} p\left(t_n \mid \mathbf{x}_n, \mathbf{w}, \beta\right)$$

where the n^{th} row of $\mathbf{X}$ is $\mathbf{x}_n^t$.

Prior:

$$p(\mathbf{w} \mid \alpha) = \prod_{i=1}^{M} \mathcal{N}\left(w_i \mid 0, \alpha_i^{-1}\right)$$

- In practice, it is difficult to integrate α out exactly.
- Instead, we use Type II Maximum Likelihood, finding ML values for each α_i.
- When we maximize the evidence with respect to these hyperparameters, many will → ∞.
- As a result, the corresponding weights will → 0, yielding a sparse solution.

RVMs for Regression

- Since both the likelihood and prior are normal, the posterior over **w** will also be normal:

Posterior:

$$p(\mathbf{w} \mid \mathbf{t}, \mathbf{X}, \alpha, \beta) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}, \Sigma)$$

where

$$\mathbf{m} = \beta \Sigma \Phi^t \mathbf{t}$$

$$\Sigma = \left(\mathbf{A} + \beta \Phi^t \Phi\right)^{-1}$$

and

$$\Phi_{ni} = \phi_i(\mathbf{x}_n)$$

$$\mathbf{A} = \mathrm{diag}(\alpha_i)$$

RVMs for Regression

- The values for α and β are determined using the evidence approximation, where we maximize

$$p(\mathbf{t} \mid \mathbf{X}, \alpha, \beta) = \int p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta) p(\mathbf{w} \mid \alpha) d\mathbf{w}$$

In general, this results in many of the precision parameters $\alpha_i \to \infty$, so that $w_i \to 0$.

Unfortunately, this is a non-convex problem.

Example