ML (cont.): SUPPORT VECTOR MACHINES

CS540 Bryan R Gibson University of Wisconsin-Madison

Slides adapted from those used by Prof. Jerry Zhu, CS540-1

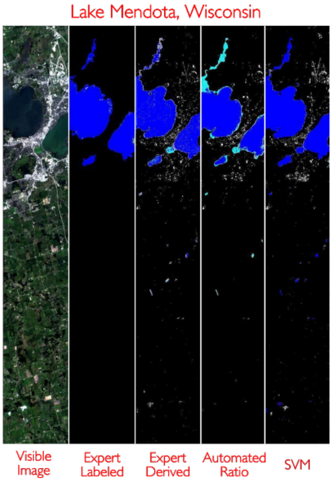
Support Vector Machines (SVMs)

The No-Math Version

Example: Lake Mendota, Madison, WI

- Identify areas of land cover (land, ice, water, snow) in a scene
- Three algorithms tried:
  - Scientist manually derived
  - Automatic best ratio
  - SVM

Classifier	Expert Derived	Automated Ratio	SVM
cloud	45.7%	43.7%	**58.5%**
ice	60.1%	34.3%	**80.4%**
land	93.6%	**94.7%**	94.0%
snow	63.5%	**90.4%**	71.6%
water	84.2%	74.3%	**89.1%**
unclassified	43.7%		

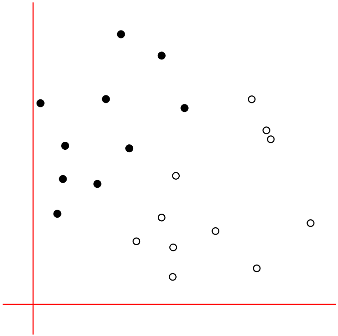


Example

- The state-of-the-art classifier

Class Labels:

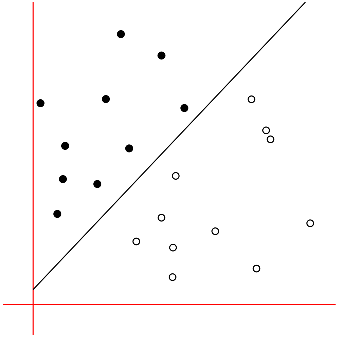
- $\bullet$ denotes $+1$
- $\circ$ denotes -1



How would you classify this data?

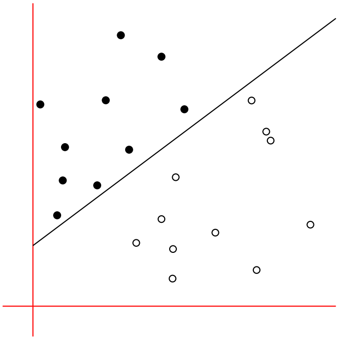
Example: Linear Classifier

- If all you can do is draw a straight line,
  or a linear decision boundary . . .



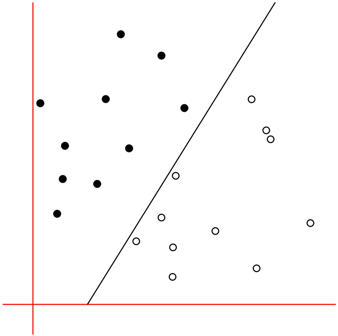
Example: Linear Classifier (cont.)

- Another "ok" decision boundary ...



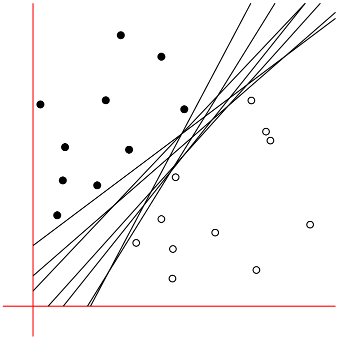
Example: Linear Classifier (cont.)

- And another . . .



Example: Linear Classifier (cont.)

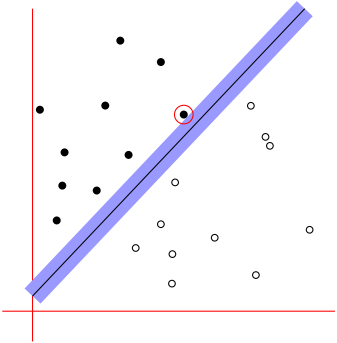
- Any of these would work ...



... but which is the best?

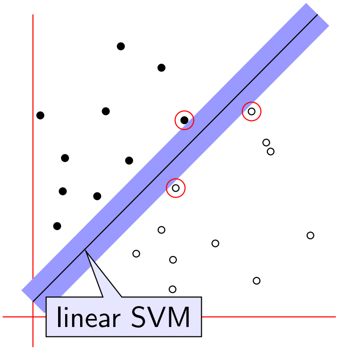
The Margin

- Margin: the width that the boundary can be increased to before hitting a data point ...



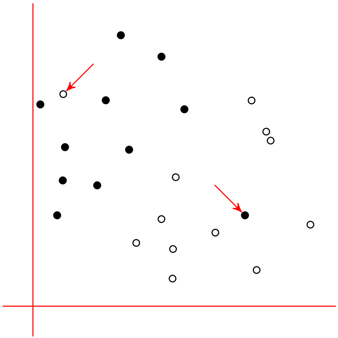
SVM: Maximize The Margin

- The simplest SVM (linear SVM) is the linear classifier with the maximum margin ...



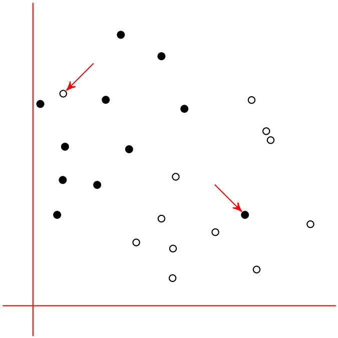
SVM: Linearly Non-Separable Data

- What if the data is not linearly separable?



SVM: Linearly Non-Separable Data (cont.)

- Two solutions:
  - Allow a few points on the wrong side (slack variables) and/or
  - Map data to higher dimensional space, classify there (kernel)



SVM: More Than 2 Classes

- N class problem: Split the task into N binary tasks
  - Class 1 vs. the rest (Class 2 to N)
  - Class 2 vs. the rest (Classes 1, 3 to N)
  - ...
  - Class N vs. the rest (Classes 1 to N-1)
- Finally, pick the class that puts the point furthest into the positive region.

SVM: Getting your hands on it

- There are many implementations
- `http://www.support-vector.net/software.html`
- `http://svmlight.joachims.org/`
- You know enough now to use SVMs

SVM: Getting your hands on it

- There are many implementations
- `http://www.support-vector.net/software.html`
- `http://svmlight.joachims.org/`
- You know enough now to use SVMs

end of lecture?

SVM: Getting your hands on it

- There are many implementations
- `http://www.support-vector.net/software.html`
- `http://svmlight.joachims.org/`

- You know enough now to use SVMs

end of lecture?

- You need to know a little more to *understand* SVMs ...

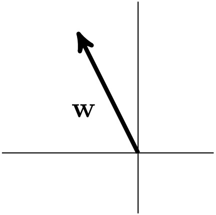
Support Vector Machines (SVMs)

The Math Version

Vectors

- A vector $\mathbf{w}$ in d-dimensional space is a list of d numbers

  e.g. $\mathbf{w} = [-1, 2]' = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$
- Written as a vertical column, $[\ldots]'$ means matrix transpose
- A vector is a line segment, with direction, in space
- The norm of a vector, written $\|\mathbf{w}\|$, is it's length



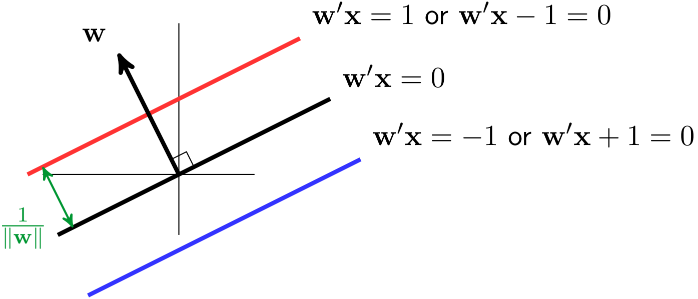
$$\|\mathbf{w}\|_2 = \sqrt{\mathbf{w}'\mathbf{w}} = \sqrt{\sum_{i=1}^{d} w_i^2}$$

$$\mathbf{x}'\mathbf{y} = \sum_i x_i y_i \text{ (inner product)}$$

What d-dimensional point $\mathbf{x}$ makes $\mathbf{w}'\mathbf{x} = 0$?

Lines

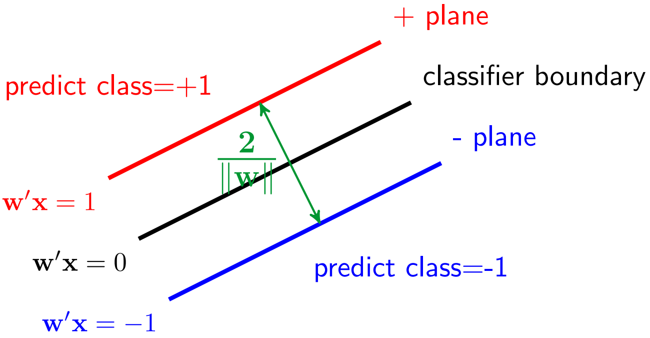
- $\mathbf{w}'\mathbf{x}$ is a scalar (single number): $\mathbf{x}$'s projection onto $\mathbf{w}$
- $\mathbf{w}'\mathbf{x} = 0$ specifies set of points $\mathbf{x}$, which is
  the line perpendicular to to $\mathbf{w}$ and intersects at $(0, 0)$



- $\mathbf{w}'\mathbf{x} = 1$ is the line parallel to $\mathbf{w}'\mathbf{x} = 0$, shifted by $\frac{1}{\|\mathbf{w}\|}$
- What if the boundary doesn't go through the origin?

SVM Boundary and Margin

- Want to find a $\mathbf{w}$ and b offset such that:
  - all positive training points $(\mathbf{x}, y = 1)$ are in red zone
  - all negative training points $(\mathbf{x}, y = -1)$ are in blue zone
  - margin m is maximized



- $m = \frac{2}{\|\mathbf{w}\|}$
- How do we find $\mathbf{w}$ and b?

SVM as Constrained Optimization

- Variables : $\mathbf{w}, b$
- Objective Function : maximize the margin $m = \frac{2}{\|\mathbf{w}\|}$

  Equiv. to **minimize** $\|\mathbf{w}\|$ or $\|\mathbf{w}\|^2 = \mathbf{w}'\mathbf{w}$ or $\frac{1}{2}\mathbf{w}'\mathbf{w}$
- Subject to each training point being on the correct side (the constraints)

- Assume n training points $(\mathbf{x}_i, y_i)_{i=1:n}$, $y \in \{-1, 1\}$
- How many constraints do we have?

SVM as Constrained Optimization (cont.)

- Variables : $\mathbf{w}, b$
- Objective Function : maximize the margin $m = \frac{2}{\|\mathbf{w}\|}$
- Subject to each training point being on the correct side (the constraints)

- Assume n training points $(\mathbf{x}_i, y_i)_{i=1:n}$, $y \in \{-1, 1\}$
- How many constraints do we have? n
  - $\mathbf{w}'\mathbf{x}_i + b \geq 1$ if $y_i = 1$
  - $\mathbf{w}'\mathbf{x}_i + b \leq -1$ if $y_i = -1$
  - Can unify as $y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1$

- We've got a continuous constrained optimization problem. What do we do?

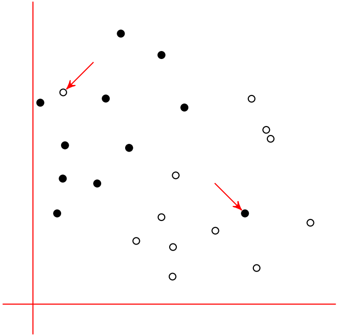
SVM as Quadratic Program (QP)

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|$$
$$\text{s.t.} \quad y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1, \quad \text{for all } i$$

- Objective is convex, quadratic
- with linear constraints
- This is known as a Quadratic Program (QP)
  for which efficient global solution algorithms exist

SVM: Non-Separable Data

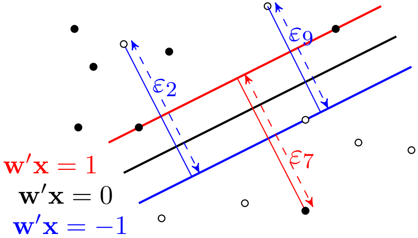
- What if the data is not linearly separable?



- Can we insist on $y_i(\mathbf{w}'\mathbf{x}_i) \geq 1, \quad$ for all i?

SVM: Non-Separable Data - Slack Variables

- Relax the contraints – allow a few "bad apples"
- For a given linear boundary $\mathbf{w}, b$ we can compute how "wrong" a bad point is by how far onto the wrong side it is (ε)



- we relax the constraints:

$$y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1 - \varepsilon_i$$

SVM: Non-Separable Data - Slack Variables (cont.)

- We want to reduce the amount of "slack" there is in the system

$$\min_{\mathbf{w},b,\varepsilon} \quad \frac{1}{2}\|\mathbf{w}\| + c\sum_i \varepsilon_i$$

$$\text{s.t.} \quad y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1 - \varepsilon_i \quad \text{for all } i$$

$$\varepsilon_i \geq 0 \quad \text{for all } i$$

- The variable c is a trade-off parameter (how to set?)
- Why do we require $\varepsilon \geq 0$?

SVM: Non-Separable Data - Slack Variables (cont.)

$$\min_{\mathbf{w},b,\varepsilon} \quad \frac{1}{2}\|\mathbf{w}\| + c\sum_i \varepsilon_i$$

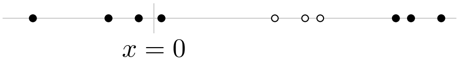
$$\text{s.t.} \quad y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1 - \varepsilon_i \quad \text{for all } i$$

$$\varepsilon_i \geq 0 \quad \text{for all } i$$

- Originally we were optimizing over $\mathbf{w}$ (d-dimensional) and b: so d+1 variables
- Now we're optimizing over ε as well: so n+d+1 variables
- With $2n$ constraints
- Still a QP: called a Soft-Margin SVM

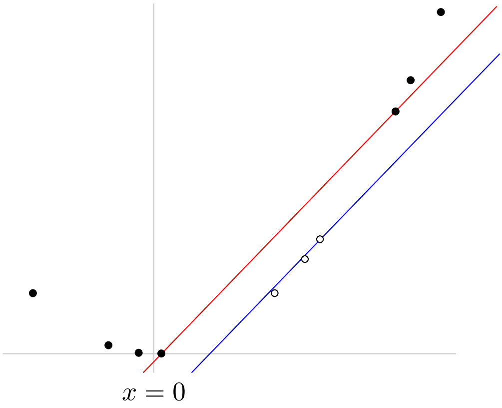
SVM: Non-Separable Data - Another Example

- Here is a non-separable dataset in 1-d space
- We could use slack variables, but instead we'll use another trick . . .



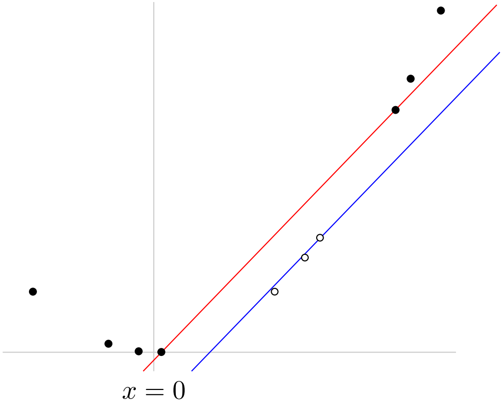
SVM: Non-Separable Data - Map to Higher Dimensions

- Let's map the data from 1-d to 2-d by $x \to (x, x^2)$



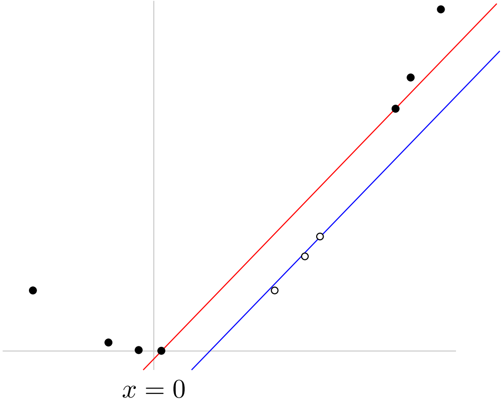
SVM: Non-Separable Data - Map to Higher Dimensions

- Let's map the data from 1-d to 2-d by $x \to (x, x^2)$
- We'll write this as $\Phi(x) = (x, x^2)$



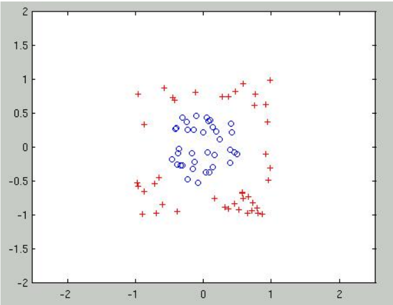
SVM: Non-Sep. Data - Map to Higher Dimensions (cont.)

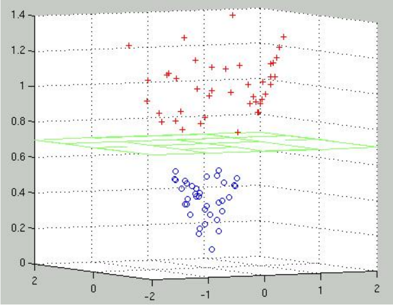
- Now the data is linearly separable in this new space!
- Can run SVM in the new space without slack
- linear boundary in new space $\rightarrow$ non-linearly boundary in old space



SVM: Non-Sep. Data - Another Example

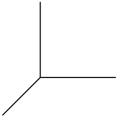
$$(x_1, x_2) \to \left(x_1, x_2, \sqrt{x_1^2 + x_2^2}\right)$$





SVM: Non-Sep. Data - Map to Higher Dimensions (cont.)

- We might want to map a high dimensional example $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ into some much higher, even infinite demensional space using $\Phi(\mathbf{x})$
- Some problems with that:
  - How do you represent infinite dimensions?
  - How do we learn (among other things) $\mathbf{w}$, which lives in this new space
  - Learning a large (or infinite) number of variables in a QP is not a good idea



$\rightarrow \infty$**?**

SVM: Non-Sep. Data - Kernels

- We'll do several things to fix this:
  - Convert into an equivalent QP problem,
    which doesn't use $\mathbf{w}$ or even $\Phi(\mathbf{x})$ alone!
    - Only uses inner product $\Phi(\mathbf{x}_i)'\Phi(\mathbf{x}_j)$
    - Solution only uses this inner product as well
    - This still seems infeasible for high (infinite) dimensions?
  - But there are smart ways to compute inner products: kernels
    - kernels are a function of two variables
    - kernel($\mathbf{x}_i$,$\mathbf{x}_j$) $\Leftrightarrow$ inner product $\Phi(\mathbf{x}_i)'\Phi(\mathbf{x}_j)$
- Why you should care:
  - Each kernel is a new (higher dim.) space
  - Fancy word to use at parties

SVM: Kernels - Biting the math bullet

- Here's the original QP problem:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}'\mathbf{w}$$
$$\text{s.t.} \quad y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1, \quad \text{for all } i$$

- Remember Lagrange multipliers?

$$L \quad = \quad \frac{1}{2}\mathbf{w}'\mathbf{w} - \sum a_i \left[y_i(\mathbf{w}'\mathbf{x}_i + b) - 1 \right]$$
$$\text{s.t.} \quad a_i \geq 1, \quad \text{for all } i$$

- New constraints due to original inequality constraints
- We want the gradient of L to vanish w.r.t. $\mathbf{w}$, b and a
- We should get:

$$\mathbf{w} = \sum a_i y_i \mathbf{x}_i$$
$$\sum a_i y_i = 0$$

and then we stick these back into the Lagrangian L ...

SVM: Kernels - Biting the math bullet (cont.)

- We get:

$$\begin{aligned} \max_{a_i} \quad & \sum a_i - \frac{1}{2} \sum_{i,j} a_i a_j y_i y_j \mathbf{x}_i' \mathbf{x}_j \\ \text{s.t.} \quad & a_i \geq 0, \quad \text{for all } i \\ & \sum a_i y_i = 0 \end{aligned}$$

- This is an equivalent QP problem (the dual)
- Before we were optimizing $\mathbf{w}$ (d variables)
  Now we optimize a (n variables)
  Which is better?
- Important: $\mathbf{x}$ only appears in the inner product!

SVM: Kernels - Biting the math bullet (cont.)

- Let's map to our new space

$$\begin{aligned} \max_{a_i} \quad & \sum a_i - \frac{1}{2}\sum_{i,j} a_i a_j y_i y_j \Phi(\mathbf{x}_i)'\Phi(\mathbf{x}_j) \\ \text{s.t.} \quad & a_i \geq 0, \quad \text{for all } i \\ & \sum a_i y_i = 0 \end{aligned}$$

- Again, this is just an inner product
- What function have we seen that we can replace this with?
- The Kernel function: $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)'\Phi(\mathbf{x}_j)$

SVM: Kernels - Biting the math bullet (cont.)

$$
\begin{aligned}
\max_{a_i} \quad & \sum a_i - \frac{1}{2}\sum_{i,j} a_i a_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\
\text{s.t.} \quad & a_i \geq 0, \quad \text{for all } i \\
& \sum a_i y_i = 0
\end{aligned}
$$

SVM: What's so special about kernels?

- Say data is 2-d: $\mathbf{s} = (s_1, s_2)$
- We decide to use a particular mapping into 6-d space:

$$\Phi(\mathbf{s}) = (s_1^2, s_2^2, \sqrt{2}s_1 s_2, \sqrt{2}s_1, \sqrt{2}s_2, 1)$$

SVM: What's so special about kernels?

- Say data is 2-d: $\mathbf{s} = (s_1, s_2)$
- We decide to use a particular mapping into 6-d space:

$$\Phi(\mathbf{s}) = (s_1^2, s_2^2, \sqrt{2}s_1 s_2, \sqrt{2}s_1, \sqrt{2}s_2, 1)$$

- Let another point be $\mathbf{t} = (t_1, t_2)$, so we get the inner product:

$$\Phi(\mathbf{s})'\Phi(\mathbf{t}) = s_1^2 t_1^2 + s_2^2 t_2^2 + 2s_1 s_2 t_1 t_2 + 2s_1 t_1 + 2s_2 t_2 + 1$$

SVM: What's so special about kernels?

- Say data is 2-d: $\mathbf{s} = (s_1, s_2)$
- We decide to use a particular mapping into 6-d space:

$$\Phi(\mathbf{s}) = (s_1^2, s_2^2, \sqrt{2}s_1 s_2, \sqrt{2}s_1, \sqrt{2}s_2, 1)$$

- Let another point be $\mathbf{t} = (t_1, t_2)$, so we get the inner product:

$$\Phi(\mathbf{s})'\Phi(\mathbf{t}) = s_1^2 t_1^2 + s_2^2 t_2^2 + 2s_1 s_2 t_1 t_2 + 2s_1 t_1 + 2s_2 t_2 + 1$$

- Let the kernel be $K(\mathbf{s}, \mathbf{t}) = (\mathbf{s}'\mathbf{t} + 1)^2$

SVM: What's so special about kernels?

- Say data is 2-d: $\mathbf{s} = (s_1, s_2)$
- We decide to use a particular mapping into 6-d space:

$$\Phi(\mathbf{s}) = (s_1^2, s_2^2, \sqrt{2}s_1 s_2, \sqrt{2}s_1, \sqrt{2}s_2, 1)$$

- Let another point be $\mathbf{t} = (t_1, t_2)$, so we get the inner product:

$$\Phi(\mathbf{s})'\Phi(\mathbf{t}) = s_1^2 t_1^2 + s_2^2 t_2^2 + 2s_1 s_2 t_1 t_2 + 2s_1 t_1 + 2s_2 t_2 + 1$$

- Let the kernel be $K(\mathbf{s}, \mathbf{t}) = (\mathbf{s}'\mathbf{t} + 1)^2$
- Verify that they're the same.
  We saved on some computation!

SVM: Choosing kernels

- "So is there a good kernel K for any Φ that I pick?"

SVM: Choosing kernels

- "So is there a good kernel K for any Φ that I pick?"
- The inverse question:
  "Given some K, is there a Φ so that $K(\mathbf{s}, \mathbf{t}) = \Phi(\mathbf{s})'\Phi(\mathbf{t})$?"

SVM: Choosing kernels

- "So is there a good kernel K for any Φ that I pick?"
- The inverse question:
  "Given some K, is there a Φ so that $K(\mathbf{s}, \mathbf{t}) = \Phi(\mathbf{s})'\Phi(\mathbf{t})$?"
- Mercer's condition: the inverse question is true ...

  if for any $g(\mathbf{s})$ such that $\int g(\mathbf{s})^2 d\mathbf{s}$ is finite we have

$$\int\int K(\mathbf{s}, \mathbf{t}) g(\mathbf{s}) g(\mathbf{t}) d\mathbf{s} d\mathbf{t} \geq 0.$$

SVM: Choosing kernels

- "So is there a good kernel K for any Φ that I pick?"
- The inverse question:
  "Given some K, is there a Φ so that $K(\mathbf{s}, \mathbf{t}) = \Phi(\mathbf{s})'\Phi(\mathbf{t})$?"
- Mercer's condition: the inverse question is true ...

  if for any $g(\mathbf{s})$ such that $\int g(\mathbf{s})^2 d\mathbf{s}$ is finite we have

  $$\int \int K(\mathbf{s}, \mathbf{t}) g(\mathbf{s}) g(\mathbf{t}) d\mathbf{s} d\mathbf{t} \geq 0.$$

- (This is positive semi-definiteness)

SVM: Choosing kernels

- "So is there a good kernel K for any Φ that I pick?"
- The inverse question:
  "Given some K, is there a Φ so that $K(\mathbf{s}, \mathbf{t}) = \Phi(\mathbf{s})'\Phi(\mathbf{t})$?"
- Mercer's condition: the inverse question is true ...

  if for any $g(\mathbf{s})$ such that $\int g(\mathbf{s})^2 d\mathbf{s}$ is finite we have

  $$\int \int K(\mathbf{s}, \mathbf{t}) g(\mathbf{s}) g(\mathbf{t}) d\mathbf{s} d\mathbf{t} \geq 0.$$

- (This is positive semi-definiteness)
- Φ may be infinite dimensional:
  we may not be able to explicitly write down Φ

SVM: Some frequently used kernels

Linear kernel: $K(\mathbf{s}, \mathbf{t}) = \mathbf{s}'\mathbf{t}$
Quadratic kernel: $K(\mathbf{s}, \mathbf{t}) = (\mathbf{s}'\mathbf{t} + 1)^2$
Polynomial kernel: $K(\mathbf{s}, \mathbf{t}) = (\mathbf{s}'\mathbf{t} + 1)^n$
Radial Basis Function kernel: $K(\mathbf{s}, \mathbf{t}) = \exp(-\|\mathbf{s} - \mathbf{t}\|^{2/\sigma})$

- ... and many, many more

- Hacking with SVM: create various kernels, hope their space Φ is meaningful, plug into SMV, pick the one with good classification accuracy (equivalent to feature engineering)
- Kernel summary:
  QP of size N, nonlinear SVM in the original space,
  new space in possibly high/infinite d,
  efficient if K is easy to compute
- Kernel can be combined with slack variables

SVM: why "support vector machine"?

- Remember, our problem can be written as:

$$
\begin{aligned}
\max_{a_i} \quad & \sum a_i - \frac{1}{2} \sum_{i,j} a_i a_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\
\text{s.t.} \quad & a_i \geq 0, \quad \text{for all } i \\
& \sum a_i y_i = 0
\end{aligned}
$$

- The decision boundary is:

$$
f(\mathbf{x}_{new}) = \mathbf{w}'\mathbf{x}_{new} + b = \sum a_i y_i \mathbf{x}_i' \mathbf{x}_{new} + b
$$

- In practice, many a's will be zero in the solution!
  - Those few $\mathbf{x}$ with $a > 0$ lie on the margins
    they are the "support vectors"

What you should know

- the intuition, and where to find the software
- Vector, line, length, norm
- Margin
- QP with linear constraints
- How to handle non-separable data
    - Slack variables
    - Kernels $\Leftrightarrow$ new feature space

- Refs:
A tutorial on Support Vector Machines for Pattern Recognition (1998) Christopher J. C. Burges
Support Vector Machines (1998) Marti A. Hearst, Intelligent Systems
An Introduction to Support Vector Machines (2000) Nello Cristianini and John Shawe-Taylor