A Gentle Introduction to Support Vector Machines in Biomedicine

Alexander Statnikov*, Douglas Hardin#, Isabelle Guyon†, Constantin F. Aliferis*

(Materials about SVM Clustering were contributed by Nikita Lytkin*)

*New York University, #Vanderbilt University, †ClopiNet

Part I

• Introduction
• Necessary mathematical concepts
• Support vector machines (SVMs) for binary classification: classical formulation
• Basic principles of statistical machine learning
Introduction

About this tutorial

• **Main goal:** *Fully understand support vector machines (and important extensions) with a modicum of mathematics knowledge.*

• This tutorial is both **modest** (*it does not invent anything new*) and **ambitious** (*support vector machines are generally considered mathematically quite difficult to grasp*).

• Our educational approach:
Data-analysis problems of interest

1. Build computational classification models (or “classifiers”) that assign objects (patients/samples) into two or more classes.
   - Classifiers can be used for diagnosis, outcome prediction, and other classification tasks.
   - E.g., build a decision-support system to diagnose primary and metastatic cancers from gene expression profiles of the patients:

```
Patient with lung cancer → Biopsy → Gene expression profile → Classifier model → Primary Lung Cancer
                       |                                           |       | Metastatic Lung Cancer
```

Data-analysis problems of interest

```
Patient → Blood sample → Mass spectrometry proteomics profile → Model
            |                                      |       |
            |                                      | Relevant to clinical trials
            |                                      | Irrelevant to clinical trials
```

```
Patient with lung cancer → Biopsy → Gene expression profile → Classifier model → Primary Lung Cancer
                       |                                           |       | Metastatic Lung Cancer
```

```
Patient → Blood sample → Mass spectrometry proteomics profile → Model
            |                                      |       |
            |                                      | Will respond to treatment
            |                                      | Will not respond to treatment
            |                                      | Cannot make decision
```
Data-analysis problems of interest

2. Build computational regression models to predict values of some continuous response variable or outcome.
   - Regression models can be used to predict survival, length of stay in the hospital, laboratory test values, etc.
   - E.g., build a decision-support system to predict optimal dosage of the drug to be administered to the patient. This dosage is determined by the values of patient biomarkers, and clinical and demographics data:

Patient → Biomarkers, clinical and demographics data → Regression model → Optimal dosage is 5 IU/Kg/week

Data-analysis problems of interest

3. Out of all measured variables in the dataset, select the smallest subset of variables that is necessary for the most accurate prediction (classification or regression) of some variable of interest (e.g., phenotypic response variable).
   - E.g., find the most compact panel of breast cancer biomarkers from microarray gene expression data for 20,000 genes:
Data-analysis problems of interest

4. Build a computational model to identify novel or outlier objects (patients/samples).
   - Such models can be used to discover deviations in sample handling protocol when doing quality control of assays, etc.
   - E.g., build a decision-support system to identify aliens.

5. Group objects (patients/samples) into several clusters based on their similarity.
   - These methods can be used to discover disease sub-types and for other tasks.
   - E.g., consider clustering of brain tumor patients into 4 clusters based on their gene expression profiles. All patients have the same pathological sub-type of the disease, and clustering discovers new molecular disease subtypes that happen to have different characteristics in terms of patient survival and time to recurrence after treatment.
Basic principles of classification

• Want to classify objects as boats and houses.

Basic principles of classification

• All objects before the coast line are boats and all objects after the coast line are houses.
• Coast line serves as a decision surface that separates two classes.
Basic principles of classification

These boats will be misclassified as houses

Basic principles of classification

This house will be misclassified as a boat
Basic principles of classification

- The methods that build classification models ("classification algorithms") operate very similarly to the previous example.
- First all objects are represented geometrically.

Basic principles of classification

Then the algorithm seeks to find a decision surface that separates classes of objects.
Basic principles of classification

The objects are classified as houses if they fall below the decision surface and as "boats" if they fall above it.

The Support Vector Machine (SVM) approach

- Support vector machines (SVMs) is a binary classification algorithm that offers a solution to problem #1.
- Extensions of the basic SVM algorithm can be applied to solve problems #1-#5.
- SVMs are important because of (a) theoretical reasons:
  - Robust to very large number of variables and small samples
  - Can learn both simple and highly complex classification models
  - Employ sophisticated mathematical principles to avoid overfitting
- (b) superior empirical results.
Main ideas of SVMs

- Consider example dataset described by 2 genes, gene X and gene Y
- Represent patients geometrically (by “vectors” or “points”)

- Find a linear decision surface (“hyperplane”) that can separate patient classes and has the largest distance (i.e., largest “gap” or “margin”) between border-line patients (i.e., “support vectors”);
Main ideas of SVMs

- If such linear decision surface does not exist, the data is mapped into a much higher dimensional space ("feature space") where the separating decision surface is found;
- The feature space is constructed via very clever mathematical projection ("kernel trick").

History of SVMs and usage in the literature

- Support vector machine classifiers have a long history of development starting from the 1950’s.
- The most important milestone for development of modern SVMs is the 1992 paper by Boser, Guyon, and Vapnik ("A training algorithm for optimal margin classifiers")
History of SVMs and usage in the literature

Use of Support Vector Machines in the Literature
- General sciences
- Biomedicine

Use of Linear Regression in the Literature
- General sciences
- Biomedicine
Necessary mathematical concepts

How to represent samples geometrically?
Vectors & points in n-dimensional space ($\mathbb{R}^n$)

- Assume that a sample/patient is described by $n$ characteristics ("features" or "variables")
- **Vector representation:** Every sample/patient is a vector in $\mathbb{R}^n$ with tail at point with 0 coordinates and arrow-head at point with the feature values.
- **Example:** Consider a patient described by 2 features:
  
  Systolic BP = 110 and Age = 29.
  
  This patient can be represented as a vector in $\mathbb{R}^2$: 

![Diagram of 2D vector space with axes for Age and Systolic BP, and a vector from (0,0) to (110,29).]
How to represent samples geometrically? Vectors & points in n-dimensional space \((\mathbb{R}^n)\)

- **Point representation:** Every sample/patient is represented as a point in the n-dimensional space \((\mathbb{R}^n)\) with coordinates given by the values of its features.

- **Example:** Consider a patient described by 2 features:
  
  \[\text{Systolic BP} = 110 \text{ and } \text{Age} = 29.\]

  This patient can be represented as a point in \(\mathbb{R}^2\):

\[
\begin{array}{c}
\text{Age} \\
\downarrow \\
(0, 0) \\
\end{array}
\begin{array}{c}
\text{Systolic BP} \\
\downarrow \\
110, 29
\end{array}
\]

### Table: Sample Representation

<table>
<thead>
<tr>
<th>Patient id</th>
<th>Cholesterol (mg/dl)</th>
<th>Systolic BP (mmHg)</th>
<th>Age (years)</th>
<th>Tail of the vector</th>
<th>Arrow-head of the vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>150</td>
<td>110</td>
<td>35</td>
<td>(0,0,0)</td>
<td>(150, 110, 35)</td>
</tr>
<tr>
<td>2</td>
<td>250</td>
<td>120</td>
<td>30</td>
<td>(0,0,0)</td>
<td>(250, 120, 30)</td>
</tr>
<tr>
<td>3</td>
<td>140</td>
<td>160</td>
<td>65</td>
<td>(0,0,0)</td>
<td>(140, 160, 65)</td>
</tr>
<tr>
<td>4</td>
<td>300</td>
<td>180</td>
<td>45</td>
<td>(0,0,0)</td>
<td>(300, 180, 45)</td>
</tr>
</tbody>
</table>

**Vector representation**

**Point representation**
Purpose of vector representation

• Having represented each sample/patient allows now to geometrically represent the decision surface that separates two groups of samples/patients.

• In order to define the decision surface, we need to introduce some basic math elements...

Basic operation on vectors in $\mathbb{R}^n$

1. Multiplication by a scalar

Consider a vector $\vec{a} = (a_1, a_2, ..., a_n)$ and a scalar $c$

Define: $c\vec{a} = (ca_1, ca_2, ..., ca_n)$

When you multiply a vector by a scalar, you “stretch” it in the same or opposite direction depending on whether the scalar is positive or negative.

$\vec{a} = (1,2)$
$c = 2$
$\vec{c}a = (2,4)$

$\vec{a} = (1,2)$
$c = -1$
$\vec{c}a = (-1, -2)$
2. Addition

Consider vectors \( \mathbf{a} = (a_1, a_2, \ldots, a_n) \) and \( \mathbf{b} = (b_1, b_2, \ldots, b_n) \)

Define: \( \mathbf{a} + \mathbf{b} = (a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n) \)

Recall addition of forces in classical mechanics.

3. Subtraction

Consider vectors \( \mathbf{a} = (a_1, a_2, \ldots, a_n) \) and \( \mathbf{b} = (b_1, b_2, \ldots, b_n) \)

Define: \( \mathbf{a} - \mathbf{b} = (a_1 - b_1, a_2 - b_2, \ldots, a_n - b_n) \)

What vector do we need to add to \( \mathbf{b} \) to get \( \mathbf{a} \)? I.e., similar to subtraction of real numbers.
Basic operation on vectors in $\mathbb{R}^n$

4. **Euclidian length or L2-norm**

Consider a vector $\vec{a} = (a_1, a_2, ..., a_n)$

Define the L2-norm: $\|\vec{a}\|_2 = \sqrt{a_1^2 + a_2^2 + \ldots + a_n^2}$

We often denote the L2-norm without subscript, i.e. $\|\vec{a}\|$

$L2$-norm is a typical way to measure length of a vector; other methods to measure length also exist.

- $\vec{a} = (1,2)$
  - $\|\vec{a}\|_2 = \sqrt{5} \approx 2.24$

5. **Dot product**

Consider vectors $\vec{a} = (a_1, a_2, ..., a_n)$ and $\vec{b} = (b_1, b_2, ..., b_n)$

Define dot product: $\vec{a} \cdot \vec{b} = a_1b_1 + a_2b_2 + \ldots + a_nb_n = \sum_{i=1}^{n} a_ib_i$

The law of cosines says that $\vec{a} \cdot \vec{b} = \|\vec{a}\| \|\vec{b}\| \cos \theta$ where $\theta$ is the angle between $\vec{a}$ and $\vec{b}$. Therefore, when the vectors are perpendicular $\vec{a} \cdot \vec{b} = 0$.

- $\vec{a} = (1,2)$
  - $\vec{b} = (3,0)$
  - $\vec{a} \cdot \vec{b} = 3$

- $\vec{a} = (0,2)$
  - $\vec{b} = (3,0)$
  - $\vec{a} \cdot \vec{b} = 0$
Basic operation on vectors in $\mathbb{R}^n$

5. Dot product (continued)

$$\vec{a} \cdot \vec{b} = a_1b_1 + a_2b_2 + \ldots + a_nb_n = \sum_{i=1}^{n} a_ib_i$$

- Property: $\vec{a} \cdot \vec{a} = a_1a_1 + a_2a_2 + \ldots + a_na_n = \|\vec{a}\|^2_2$
- In the classical regression equation $y = \vec{w} \cdot \vec{x} + b$

the response variable $y$ is just a dot product of the vector representing patient characteristics ($\vec{x}$) and the regression weights vector ($\vec{w}$) which is common across all patients plus an offset $b$.

Hyperplanes as decision surfaces

- A hyperplane is a linear decision surface that splits the space into two parts;
- It is obvious that a hyperplane is a binary classifier.

A hyperplane in $\mathbb{R}^2$ is a line

A hyperplane in $\mathbb{R}^3$ is a plane

A hyperplane in $\mathbb{R}^n$ is an $n-1$ dimensional subspace
Equation of a hyperplane

First we show with show the definition of hyperplane by an interactive demonstration.

or go to http://www.dsl-lab.org/svm_tutorial/planedemo.html

Source: http://www.math.umn.edu/~nykamp/

Equation of a hyperplane

Consider the case of $\mathbb{R}^3$:

![Diagram of hyperplane]

An equation of a hyperplane is defined by a point $(P_0)$ and a perpendicular vector to the plane ($\vec{w}$) at that point.

Define vectors: $\vec{x}_0 = \overrightarrow{OP_0}$ and $\vec{x} = \overrightarrow{OP}$, where $P$ is an arbitrary point on a hyperplane.

A condition for $P$ to be on the plane is that the vector $\vec{x} - \vec{x}_0$ is perpendicular to $\vec{w}$:

$$\vec{w} \cdot (\vec{x} - \vec{x}_0) = 0$$

or

$$\vec{w} \cdot \vec{x} - \vec{w} \cdot \vec{x}_0 = 0$$

define $b = -\vec{w} \cdot \vec{x}_0$

$$\vec{w} \cdot \vec{x} + b = 0$$

The above equations also hold for $\mathbb{R}^n$ when $n>3$. 
**Equation of a hyperplane**

**Example**

\( \vec{w} = (4, -1, 6) \)

\( P_0 = (0, 1, -7) \)

\( b = -\vec{w} \cdot \vec{x}_0 = -(0 - 1 - 42) = 43 \)

\[ \Rightarrow \vec{w} \cdot \vec{x} + 43 = 0 \]

\[ \Rightarrow (4, -1, 6) \cdot \vec{x} + 43 = 0 \]

\[ \Rightarrow (4, -1, 6) \cdot (x_1, x_2, x_3) + 43 = 0 \]

\[ \Rightarrow 4x_1 - x_2 + 6x_3 + 43 = 0 \]

What happens if the \( b \) coefficient changes?
The hyperplane moves along the direction of \( \vec{w} \).
We obtain “parallel hyperplanes”.

Distance between two parallel hyperplanes \( \vec{w} \cdot \vec{x} + b_1 = 0 \) and \( \vec{w} \cdot \vec{x} + b_2 = 0 \)
is equal to \( D = \left| b_1 - b_2 \right| / \| \vec{w} \| \).

**(Derivation of the distance between two parallel hyperplanes)**

\( \vec{x}_2 = \vec{x}_1 + t\vec{w} \)

\[ D = \| t\vec{w} \| = \| t \| \| \vec{w} \| \]

\( \vec{w} \cdot \vec{x}_2 + b_2 = 0 \)

\( \vec{w} \cdot (\vec{x}_1 + t\vec{w}) + b_2 = 0 \)

\( \vec{w} \cdot \vec{x}_1 + t\| \vec{w} \|^2 + b_2 = 0 \)

\( (\vec{w} \cdot \vec{x}_1 + b_1) - b_1 + t\| \vec{w} \|^2 + b_2 = 0 \)

\[ -b_1 + t\| \vec{w} \|^2 + b_2 = 0 \]

\[ t = (b_1 - b_2) / \| \vec{w} \|^2 \]

\[ \Rightarrow D = \| t \| \| \vec{w} \| = \left| b_1 - b_2 \right| / \| \vec{w} \| \]
Recap

We know...
• How to represent patients (as “vectors”)
• How to define a linear decision surface (“hyperplane”)

We need to know...
• How to efficiently compute the hyperplane that separates two classes with the largest “gap”?

Need to introduce basics of relevant optimization theory

Basics of optimization: Convex functions

• A function is called convex if the function lies below the straight line segment connecting two points, for any two points in the interval.
• Property: Any local minimum is a global minimum!
Basics of optimization: Quadratic programming (QP)

- Quadratic programming (QP) is a special optimization problem: the function to optimize ("objective") is quadratic, subject to linear constraints.
- Convex QP problems have convex objective functions.
- These problems can be solved easily and efficiently by greedy algorithms (because every local minimum is a global minimum).

Basics of optimization: Example QP problem

Consider $\bar{x} = (x_1, x_2)$

Minimize $\frac{1}{2} \| \bar{x} \|_2^2$ subject to $x_1 + x_2 - 1 \geq 0$

This is QP problem, and it is a convex QP as we will see later.

We can rewrite it as:

Minimize $\frac{1}{2} (x_1^2 + x_2^2)$ subject to $x_1 + x_2 - 1 \geq 0$
Basics of optimization: Example QP problem

\[ f(x_1, x_2) \quad x_1 + x_2 - 1 \geq 0 \]
\[ \frac{1}{2} (x_1^2 + x_2^2) \quad x_1 + x_2 - 1 \leq 0 \]

The solution is \( x_1 = 1/2 \) and \( x_2 = 1/2 \).

Congratulations! You have mastered all math elements needed to understand support vector machines.

Now, let us strengthen your knowledge by a quiz 😊
Quiz

1) Consider a hyperplane shown with white. It is defined by equation: \( \vec{w} \cdot \vec{x} + 10 = 0 \)
   Which of the three other hyperplanes can be defined by equation: \( \vec{w} \cdot \vec{x} + 3 = 0 \)?
   - Orange
   - Green
   - Yellow

2) What is the dot product between vectors \( \vec{a} = (3,3) \) and \( \vec{b} = (1,-1) \)?

Quiz

3) What is the dot product between vectors \( \vec{a} = (3,3) \) and \( \vec{b} = (1,0) \)?

4) What is the length of a vector \( \vec{a} = (2,0) \) and what is the length of all other red vectors in the figure?
5) Which of the four functions is/are convex?

1

2

3

4

Support vector machines (SVMs) for binary classification: classical formulation
Case 1: Linearly separable data; “Hard-margin” linear SVM

Given training data:
\[ \bar{x}_1, \bar{x}_2, \ldots, \bar{x}_N \in \mathbb{R}^n \]
\[ y_1, y_2, \ldots, y_N \in \{-1, +1\} \]

- Want to find a classifier (hyperplane) to separate negative objects from the positive ones.
- An infinite number of such hyperplanes exist.
- SVMs finds the hyperplane that maximizes the gap between data points on the boundaries (so-called “support vectors”).
- If the points on the boundaries are not informative (e.g., due to noise), SVMs may not do well.

Statement of linear SVM classifier

The gap is distance between parallel hyperplanes:
\[ \bar{w} \cdot \bar{x} + b = -1 \quad \text{and} \quad \bar{w} \cdot \bar{x} + b = +1 \]

Or equivalently:
\[ \bar{w} \cdot \bar{x} + (b + 1) = 0 \]
\[ \bar{w} \cdot \bar{x} + (b - 1) = 0 \]

We know that
\[ D = |b_1 - b_2| / ||\bar{w}|| \]

Therefore:
\[ D = 2 / ||\bar{w}|| \]

Since we want to maximize the gap,
we need to minimize \( ||\bar{w}|| \)

or equivalently minimize \( \frac{1}{2} ||\bar{w}||^2 \) (\( \frac{1}{2} \) is convenient for taking derivative later on)
**Statement of linear SVM classifier**

In addition we need to impose constraints that all objects are correctly classified. In our case:

\[ \mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \quad \text{if} \quad y_i = -1 \]
\[ \mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \quad \text{if} \quad y_i = +1 \]

Equivalently:
\[ y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \]

In summary:

Want to minimize \( \frac{1}{2} \| \mathbf{w} \|^2 \) subject to \( y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \) for \( i = 1, \ldots, N \)

Then given a new object \( \mathbf{x} \), the classifier is \( f(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) \)

---

**SVM optimization problem:**

**Primal formulation**

Minimize \( \frac{1}{2} \sum_{i=1}^{n} w_i^2 \) subject to \( y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \) for \( i = 1, \ldots, N \)

- This is called “primal formulation of linear SVMs”.
- It is a convex quadratic programming (QP) optimization problem with \( n \) variables \( (w_i, i = 1, \ldots, n) \), where \( n \) is the number of features in the dataset.
SVM optimization problem: Dual formulation

• The previous problem can be recast in the so-called “dual form” giving rise to “dual formulation of linear SVMs”.
• It is also a convex quadratic programming problem but with $N$ variables ($\alpha_i, i = 1, ..., N$), where $N$ is the number of samples.

Maximize $\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \overrightarrow{x}_i \cdot \overrightarrow{x}_j$ subject to $\alpha_i \geq 0$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$.

Then the $w$-vector is defined in terms of $\alpha_i$: $\bar{w} = \sum_{i=1}^{N} \alpha_i y_i \overrightarrow{x}_i$

And the solution becomes: $f(\overrightarrow{x}) = \text{sign}(\sum_{i=1}^{N} \alpha_i y_i \overrightarrow{x}_i \cdot \overrightarrow{x} + b)$

SVM optimization problem: Benefits of using dual formulation

1) No need to access original data, need to access only dot products.

Objective function: $\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \overrightarrow{x}_i \cdot \overrightarrow{x}_j$

Solution: $f(\overrightarrow{x}) = \text{sign}(\sum_{i=1}^{N} \alpha_i y_i \overrightarrow{x}_i \cdot \overrightarrow{x} + b)$

2) Number of free parameters is bounded by the number of support vectors and not by the number of variables (beneficial for high-dimensional problems!).

E.g., if a microarray dataset contains 20,000 genes and 100 patients, then need to find only up to 100 parameters!
Minimize $\frac{1}{2} \sum_{i=1}^{n} w_i^2$ subject to $y_i (\tilde{w} \cdot \tilde{x}_i + b) - 1 \geq 0$ for $i = 1, \ldots, N$

Objective function

Constraints

Apply the method of Lagrange multipliers.

Define Lagrangian $\Lambda_p (\tilde{w}, b, \alpha) = \frac{1}{2} \sum_{i=1}^{n} w_i^2 - \sum_{i=1}^{N} \alpha_i (y_i (\tilde{w} \cdot \tilde{x}_i + b) - 1)$

We need to minimize this Lagrangian with respect to $\tilde{w}, b$ and simultaneously require that the derivative with respect to $\alpha$ vanishes, all subject to the constraints that $\alpha_i \geq 0$.

If we set the derivatives with respect to $\tilde{w}, b$ to 0, we obtain:

$$\frac{\partial \Lambda_p (\tilde{w}, b, \alpha)}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0$$
$$\frac{\partial \Lambda_p (\tilde{w}, b, \alpha)}{\partial \tilde{w}} = 0 \Rightarrow \tilde{w} = \sum_{i=1}^{N} \alpha_i y_i \tilde{x}_i$$

We substitute the above into the equation for $\Lambda_p (\tilde{w}, b, \alpha)$ and obtain “dual formulation of linear SVMs”:

$$\Lambda_D (\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \tilde{x}_i \cdot \tilde{x}_j$$

We seek to maximize the above Lagrangian with respect to $\alpha$, subject to the constraints that $\alpha_i \geq 0$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$.
Case 2: Not linearly separable data; “Soft-margin” linear SVM

What if the data is not linearly separable? E.g., there are outliers or noisy measurements, or the data is slightly non-linear.

Want to handle this case without changing the family of decision functions.

Approach:

Assign a “slack variable” to each object $\xi_i \geq 0$, which can be thought of distance from the separating hyperplane if an object is misclassified and 0 otherwise.

Want to minimize $\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \xi_i$ subject to $y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i$ for $i = 1, \ldots, N$

Then given a new object $x$, the classifier is $f(x) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$

Two formulations of soft-margin linear SVM

Primal formulation:

Minimize $\frac{1}{2} \sum_{i=1}^{n} w_i^2 + C \sum_{i=1}^{N} \xi_i$ subject to $y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i$ for $i = 1, \ldots, N$

Dual formulation:

Minimize $\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$ subject to $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$

for $i = 1, \ldots, N$. 

Objective function Constraints

Objective function Constraints
**Parameter C in soft-margin SVM**

Minimize $\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \xi_i$ subject to $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i$ for $i = 1, \ldots, N$

- When $C$ is very large, the soft-margin SVM is equivalent to hard-margin SVM;
- When $C$ is very small, we admit misclassifications in the training data at the expense of having $w$-vector with small norm;
- $C$ has to be selected for the distribution at hand as it will be discussed later in this tutorial.

**Case 3: Not linearly separable data; Kernel trick**

Data is not linearly separable in the input space

Data is linearly separable in the feature space obtained by a kernel

$\Phi: \mathbb{R}^N \rightarrow \mathcal{H}$
Kernel trick

Original data $\tilde{x}$ (in input space)

$$f(x) = \text{sign}(\tilde{w} \cdot \tilde{x} + b)$$

$$\tilde{w} = \sum_{i=1}^{N} \alpha_i y_i \tilde{x}_i$$

Data in a higher dimensional feature space $\Phi(\tilde{x})$

$$f(x) = \text{sign}(\tilde{w} \cdot \Phi(\tilde{x}) + b)$$

$$\tilde{w} = \sum_{i=1}^{N} \alpha_i y_i \Phi(\tilde{x}_i)$$

$$f(x) = \text{sign}(\sum_{i=1}^{N} \alpha_i y_i \Phi(\tilde{x}_i) \cdot \Phi(\tilde{x}) + b)$$

$$f(x) = \text{sign}(\sum_{i=1}^{N} \alpha_i y_i K(\tilde{x}_i, \tilde{x}) + b)$$

Therefore, we do not need to know $\Phi$ explicitly, we just need to define function $K(\cdot, \cdot) : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$.

Not every function $\mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$ can be a valid kernel; it has to satisfy so-called Mercer conditions. Otherwise, the underlying quadratic program may not be solvable.

Popular kernels

A kernel is a dot product in some feature space:

$$K(\tilde{x}_i, \tilde{x}_j) = \Phi(\tilde{x}_i) \cdot \Phi(\tilde{x}_j)$$

Examples:

- $K(\tilde{x}_i, \tilde{x}_j) = \tilde{x}_i \cdot \tilde{x}_j$  
  Linear kernel

- $K(\tilde{x}_i, \tilde{x}_j) = \exp(-\gamma \|\tilde{x}_i - \tilde{x}_j\|^2)$  
  Gaussian kernel

- $K(\tilde{x}_i, \tilde{x}_j) = \exp(-\gamma \|\tilde{x}_i - \tilde{x}_j\|)$  
  Exponential kernel

- $K(\tilde{x}_i, \tilde{x}_j) = (p + \tilde{x}_i \cdot \tilde{x}_j)^q$  
  Polynomial kernel

- $K(\tilde{x}_i, \tilde{x}_j) = (p + \tilde{x}_i \cdot \tilde{x}_j)^q \exp(-\gamma \|\tilde{x}_i - \tilde{x}_j\|^2)$  
  Hybrid kernel

- $K(\tilde{x}_i, \tilde{x}_j) = \text{tanh}(k\tilde{x}_i \cdot \tilde{x}_j - \delta)$  
  Sigmoidal
Understanding the Gaussian kernel

Consider Gaussian kernel: \( K(\tilde{x}, \tilde{x}_j) = \exp(-\gamma \|\tilde{x} - \tilde{x}_j\|^2) \)

- All data points in the input space are mapped to a multi-dimensional sphere.
- The parameter \( \gamma \) defines the relative location of the points on the sphere.

Understanding the polynomial kernel

Consider polynomial kernel: \( K(\tilde{x}_i, \tilde{x}_j) = (1 + \tilde{x}_i \cdot \tilde{x}_j)^3 \)

Assume that we are dealing with 2-dimensional data (i.e., in \( \mathbb{R}^2 \)). Where will this kernel map the data?

2-dimensional space

\[ x_{(1)} \quad x_{(2)} \]

kernel

10-dimensional space

\[ 1 \quad x_{(1)} \quad x_{(2)} \quad x_{(1)}^2 \quad x_{(2)}^2 \quad x_{(1)}x_{(2)} \quad x_{(1)}^3 \quad x_{(2)}^3 \quad x_{(1)}x_{(2)}^2 \quad x_{(1)}^2x_{(2)} \]
Example of benefits of using a kernel

- Data is not linearly separable in the input space ($\mathbb{R}^2$).
- Apply kernel $K(\tilde{x}, \tilde{z}) = (\tilde{x} \cdot \tilde{z})^2$ to map data to a higher dimensional space (3-dimensional) where it is linearly separable.

$$K(\tilde{x}, \tilde{z}) = (\tilde{x} \cdot \tilde{z})^2 = \left[ \begin{pmatrix} x(1) \\ x(2) \end{pmatrix}, \begin{pmatrix} z(1) \\ z(2) \end{pmatrix} \right]^2 = \begin{pmatrix} x(1)z(1) + x(2)z(2) \end{pmatrix}^2 =$$

$$= x(1)^2z(1)^2 + 2x(1)z(1)x(2)z(2) + x(2)^2z(2)^2 = \begin{pmatrix} \sqrt{2}x(1)x(2) \\ x(2)^2 \end{pmatrix} \cdot \begin{pmatrix} \sqrt{2}z(1)z(2) \\ z(2)^2 \end{pmatrix} = \Phi(\tilde{x}) \cdot \Phi(\tilde{z})$$

Therefore, the explicit mapping is $\Phi(\tilde{x}) = \begin{pmatrix} x(1)^2 \\ \sqrt{2}x(1)x(2) \\ x(2)^2 \end{pmatrix}$
Comparison with methods from classical statistics & regression

• Need ≥ 5 samples for each parameter of the regression model to be estimated:

<table>
<thead>
<tr>
<th>Number of variables</th>
<th>Polynomial degree</th>
<th>Number of parameters</th>
<th>Required sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>286</td>
<td>1,430</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>3,003</td>
<td>15,015</td>
</tr>
<tr>
<td>100</td>
<td>3</td>
<td>176,851</td>
<td>884,255</td>
</tr>
<tr>
<td>100</td>
<td>5</td>
<td>96,560,646</td>
<td>482,803,230</td>
</tr>
</tbody>
</table>

• SVMs do not have such requirement & often require much less sample than the number of variables, even when a high-degree polynomial kernel is used.

Basic principles of statistical machine learning
Generalization and overfitting

- **Generalization**: A classifier or a regression algorithm learns to correctly predict output from given inputs not only in previously seen objects but also in previously unseen objects.

- **Overfitting**: A classifier or a regression algorithm learns to correctly predict output from given inputs in previously seen objects but fails to do so in previously unseen objects.

- **Overfitting** → Poor generalization.

Example of overfitting and generalization

There is a linear relationship between predictor and outcome (plus some Gaussian noise).

- Algorithm 1 learned non-reproducible peculiarities of the specific sample available for learning but did not learn the general characteristics of the function that generated the data. Thus, it is overfitted and has poor generalization.

- Algorithm 2 learned general characteristics of the function that produced the data. Thus, it generalizes.
Example of overfitting and generalization

A general principle here (Occam’s razor) is that the likelihood of overfitting increases with the “complexity” of an algorithm.

“Loss + penalty” paradigm for learning to avoid overfitting and ensure generalization

- Many statistical learning algorithms (including SVMs) search for a decision function by solving the following optimization problem:

  Minimize \( (Loss + \lambda \text{Penalty}) \)

- \( Loss \) measures error of fitting the data
- \( \text{Penalty} \) penalizes complexity of the learned function
- \( \lambda \) is regularization parameter that balances \( Loss \) and \( \text{Penalty} \)
“Loss + penalty” paradigm for learning to avoid overfitting and ensure generalization

The classifier that is represented by a straight line is considered simpler than the classifier represented by a wiggled line (because we need less information to represent it).

SVMs in “loss + penalty” form

SVMs build the following classifiers: $f(\bar{x}) = \text{sign}(\bar{w} \cdot \bar{x} + b)$

Consider soft-margin linear SVM formulation:

Find $\bar{w}$ and $b$ that

Minimize $\frac{1}{2} \|\bar{w}\|^2 + C \sum_{i=1}^{N} \xi_i$ subject to $y_i (\bar{w} \cdot \bar{x}_i + b) \geq 1 - \xi_i$ for $i = 1, \ldots, N$

This can also be stated as:

Find $\bar{w}$ and $b$ that

Minimize $\sum_{i=1}^{N} [1 - y_i f(\bar{x}_i)]_+ + \lambda \|\bar{w}\|_2^2$

Loss Penalty

("hinge loss")

(in fact, one can show that $\lambda = 1/(2C)$).
Meaning of SVM loss function

Consider loss function: \( \sum_{i=1}^{N} [1 - y_i f(\tilde{x}_i)]_+ \)

- Recall that \([...]_+\) indicates the positive part
- For a given sample/patient \(i\), the loss is non-zero if \(1 - y_i f(\tilde{x}_i) > 0\)
- In other words, \(y_i f(\tilde{x}_i) < 1\)
- Since \(y_i = \{-1, +1\}\), this means that the loss is non-zero if
  \(f(\tilde{x}_i) < 1\) for \(y_i = +1\)
  \(f(\tilde{x}_i) > -1\) for \(y_i = -1\)
- In other words, the loss is non-zero if
  \(\tilde{w} \cdot \tilde{x}_i + b < 1\) for \(y_i = +1\)
  \(\tilde{w} \cdot \tilde{x}_i + b > -1\) for \(y_i = -1\)

Meaning of SVM loss function

- If the object is negative, it is penalized only in regions 2,3,4
- If the object is positive, it is penalized only in regions 1,2,3
Flexibility of “loss + penalty” framework

Minimize \((\text{Loss} + \lambda \text{Penalty})\)

<table>
<thead>
<tr>
<th>Loss function</th>
<th>Penalty function</th>
<th>Resulting algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hinge loss: (\sum_{i=1}^{N}[1 - y_i(\mathbf{w} \cdot \mathbf{x}<em>i + b)]</em>+)</td>
<td>(2</td>
<td></td>
</tr>
<tr>
<td>Hinge loss: (\sum_{i=1}^{N}[1 - y_i(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}<em>i) + b)]</em>+)</td>
<td>(2</td>
<td></td>
</tr>
<tr>
<td>Hinge loss: (\sum_{i=1}^{N}[1 - y_i(\mathbf{w} \cdot \mathbf{x}<em>i + b)]</em>+)</td>
<td>(2</td>
<td></td>
</tr>
<tr>
<td>Hinge loss: (\sum_{i=1}^{N}[1 - y_i(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}<em>i) + b)]</em>+)</td>
<td>(2</td>
<td></td>
</tr>
<tr>
<td>Squared error: (\sum_{i=1}^{N}(y_i - (\mathbf{w} \cdot \mathbf{x}_i + b))^2)</td>
<td>(-)</td>
<td>Multiple Linear Regression</td>
</tr>
<tr>
<td>Squared error: (\sum_{i=1}^{N}(y_i - (\mathbf{w} \cdot \mathbf{x}_i + b))^2)</td>
<td>(2</td>
<td></td>
</tr>
<tr>
<td>Squared error: (\sum_{i=1}^{N}(y_i - (\mathbf{w} \cdot \mathbf{x}_i + b))^2)</td>
<td>(2</td>
<td></td>
</tr>
<tr>
<td>Squared error: (\sum_{i=1}^{N}(y_i - (\mathbf{w} \cdot \mathbf{x}_i + b))^2)</td>
<td>(\lambda_1</td>
<td></td>
</tr>
<tr>
<td>Negative log-likelihood: (\sum_{i=1}^{N}\log(1 + \exp^{-\gamma(y_i, k)}))</td>
<td>(2</td>
<td></td>
</tr>
<tr>
<td>Negative log-likelihood: (\sum_{i=1}^{N}\log(1 + \exp^{-\gamma(y_i, k) + \gamma}))</td>
<td>(2</td>
<td></td>
</tr>
</tbody>
</table>

Part 2

- Model selection for SVMs
- Extensions to the basic SVM model:
  1. SVMs for multiclass data
  2. Support vector regression
  3. Novelty detection with SVM-based methods
  4. Support vector clustering
  5. SVM-based variable selection
  6. Computing posterior class probabilities for SVM classifiers
Model selection for SVMs

Need for model selection for SVMs

- It is impossible to find a linear SVM classifier that separates tumors from normals!
- Need a non-linear SVM classifier, e.g. SVM with polynomial kernel of degree 2 solves this problem without errors.

- We should not apply a non-linear SVM classifier while we can perfectly solve this problem using a linear SVM classifier!
A data-driven approach for model selection for SVMs

• Do not know *a priori* what type of SVM kernel and what kernel parameter(s) to use for a given dataset?
• Need to examine various combinations of parameters, e.g. consider searching the following grid:

<table>
<thead>
<tr>
<th>Parameter C</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.1, 1)</td>
</tr>
<tr>
<td>(1, 1)</td>
</tr>
<tr>
<td>(10, 1)</td>
</tr>
<tr>
<td>(100, 1)</td>
</tr>
<tr>
<td>(1000, 1)</td>
</tr>
<tr>
<td>(0.1, 2)</td>
</tr>
<tr>
<td>(1, 2)</td>
</tr>
<tr>
<td>(10, 2)</td>
</tr>
<tr>
<td>(100, 2)</td>
</tr>
<tr>
<td>(1000, 2)</td>
</tr>
<tr>
<td>(0.1, 3)</td>
</tr>
<tr>
<td>(1, 3)</td>
</tr>
<tr>
<td>(10, 3)</td>
</tr>
<tr>
<td>(100, 3)</td>
</tr>
<tr>
<td>(1000, 3)</td>
</tr>
<tr>
<td>(0.1, 4)</td>
</tr>
<tr>
<td>(1, 4)</td>
</tr>
<tr>
<td>(10, 4)</td>
</tr>
<tr>
<td>(100, 4)</td>
</tr>
<tr>
<td>(1000, 4)</td>
</tr>
<tr>
<td>(0.1, 5)</td>
</tr>
<tr>
<td>(1, 5)</td>
</tr>
<tr>
<td>(10, 5)</td>
</tr>
<tr>
<td>(100, 5)</td>
</tr>
<tr>
<td>(1000, 5)</td>
</tr>
</tbody>
</table>

• How to search this grid while producing an unbiased estimate of classification performance?

**Nested cross-validation**

Recall the main idea of cross-validation:

What combination of SVM parameters to apply on training data?

Perform "grid search" using another nested loop of cross-validation.
Example of nested cross-validation

Consider that we use 3-fold cross-validation and we want to optimize parameter C that takes values “1” and “2”.

### Outer Loop

<table>
<thead>
<tr>
<th>Training set</th>
<th>Testing set</th>
<th>C</th>
<th>Accuracy</th>
<th>Average Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1, P2</td>
<td>P3</td>
<td>1</td>
<td>89%</td>
<td>83%</td>
</tr>
<tr>
<td>P1, P3</td>
<td>P2</td>
<td>2</td>
<td>84%</td>
<td></td>
</tr>
<tr>
<td>P2, P3</td>
<td>P1</td>
<td>1</td>
<td>76%</td>
<td></td>
</tr>
</tbody>
</table>

### Inner Loop

<table>
<thead>
<tr>
<th>Training set</th>
<th>Validation set</th>
<th>C</th>
<th>Accuracy</th>
<th>Average Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>P2</td>
<td>1</td>
<td>86%</td>
<td>85%</td>
</tr>
<tr>
<td>P2</td>
<td>P1</td>
<td>2</td>
<td>84%</td>
<td></td>
</tr>
<tr>
<td>P1</td>
<td>P2</td>
<td>2</td>
<td>70%</td>
<td></td>
</tr>
<tr>
<td>P2</td>
<td>P1</td>
<td>1</td>
<td>90%</td>
<td>80%</td>
</tr>
</tbody>
</table>

On use of cross-validation

- Empirically we found that cross-validation works well for model selection for SVMs in many biomedical problem domains;
- Many other approaches that can be used for model selection for SVMs exist, e.g.:
  - Virtual leave-one-out cross-validation
  - Generalized cross-validation
  - Bayesian information criterion (BIC)
  - Minimum description length (MDL)
  - Vapnik-Chernovenkis (VC) dimension
  - Bootstrap
SVMs for multiclass data

One-versus-rest multiclass SVM method
One-versus-one multicategory SVM method

DAGSVM multicategory SVM method
SVM multicategory methods by Weston and Watkins and by Crammer and Singer

Support vector regression
**ε-Support vector regression (ε-SVR)**

Given training data:
\[
\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_N \in \mathbb{R}^n \\
y_1, y_2, \ldots, y_N \in \mathbb{R}
\]

**Main idea:**
Find a function \( f(\bar{x}) = \bar{w} \cdot \bar{x} + b \) that approximates \( y_1, \ldots, y_N \):
- it has at most ε derivation from the true values \( y_i \)
- it is as “flat” as possible (to avoid overfitting)

E.g., build a model to predict survival of cancer patients that can admit a one month error (= ε).
Formulation of “hard-margin” $\varepsilon$-SVR

Find $f(\bar{x}) = \vec{w} \cdot \bar{x} + b$ by minimizing $\frac{1}{2} \| \vec{w} \|^2$ subject to constraints:

$$y_i - (\vec{w} \cdot \bar{x} + b) \leq \varepsilon$$
$$y_i - (\vec{w} \cdot \bar{x} + b) \geq -\varepsilon$$

for $i = 1, \ldots, N$.

I.e., difference between $y_i$ and the fitted function should be smaller than $\varepsilon$ and larger than $-\varepsilon$ ⇔ the fitted function should go through $\varepsilon$-neighborhood of all points $y_i$.

Influence of parameter $\varepsilon$

$y = f(x)$

$\varepsilon = 0.05$  $\varepsilon = 0.10$  $\varepsilon = 0.15$

$\varepsilon = 0.20$  $\varepsilon = 0.25$  $\varepsilon = 0.30$
Formulation of “soft-margin” $\varepsilon$-SVR

If we have points like this (e.g., outliers or noise) we can either:

a) increase $\varepsilon$ to ensure that these points are within the new $\varepsilon$-neighborhood, or

b) assign a penalty (“slack” variable) to each of this points (as was done for “soft-margin” SVMs)

Find $f(\bar{x}) = \vec{w} \cdot \bar{x} + b$
by minimizing $\frac{1}{2} \| \vec{w} \|^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*)$
subject to constraints:

$y_i - (\vec{w} \cdot \bar{x} + b) \leq \varepsilon + \xi_i$
$y_i - (\vec{w} \cdot \bar{x} + b) \geq -\varepsilon - \xi_i^*$
$\xi_i, \xi_i^* \geq 0$
for $i = 1, \ldots, N.$

• $\xi_i = 0$ if the linear function is above the lower boundary of the $\varepsilon$-neighborhood; otherwise it denotes distance from the linear function to the lower boundary.

• $\xi_i^* = 0$ if the linear function is below the upper boundary of the $\varepsilon$-neighborhood; otherwise it denotes distance from the linear function to the upper boundary.

• Notice that only points outside $\varepsilon$-neighborhood are penalized!
Nonlinear $\varepsilon$-SVR

$y = f(x)$  
**Input space**

$\Phi(x)$  
**Feature space**

$\Phi(\bar{x}_i)$  
Linear relation does not exist

$\Phi^{-1}(\bar{x}_i)$  
Corresponding non-linear relation

$\varepsilon$-Support vector regression in “loss + penalty” form

Build decision function of the form: $f(\bar{x}) = \bar{w} \cdot \bar{x} + b$

Find $\bar{w}$ and $b$ that

Minimize $\sum_{i=1}^{N} \max(0, |y_i - f(\bar{x}_i)| - \varepsilon) + \lambda \|\bar{w}\|_2^2$

$Loss$  
($“linear \ \varepsilon$-insensitive loss”)

$Penalty$

Loss function value

$\zeta$

$-\varepsilon \quad +\varepsilon$  
Error in approximation
Comparing loss functions of regression methods

### Linear \(\epsilon\)-insensitive loss

- Loss function: \(\sum_{i=1}^{n} \max(0, |y_i - (\mathbf{w} \cdot \mathbf{x}_i + b) - \epsilon)|\)
- Penalty function: \(\lambda |\mathbf{w}|^p\)
- Resulting algorithm: \(\epsilon\)-insensitive linear SVR

### Quadratic \(\epsilon\)-insensitive loss

- Loss function: \(\sum_{i=1}^{n} \max(0, |y_i - (\mathbf{w} \cdot \Phi(\mathbf{c}_i) + b) - \epsilon)|^2\)
- Penalty function: \(\lambda |\mathbf{w}|^p\)
- Resulting algorithm: \(\epsilon\)-insensitive non-linear SVR

### Mean squared error

- Loss function: \(\sum_{i=1}^{n} (y_i - (\mathbf{w} \cdot \mathbf{x}_i + b))^2\)
- Penalty function: \(-\)
- Resulting algorithm: Multiple Linear Regression (least squares formulation)

### Mean linear error

- Loss function: \(\sum_{i=1}^{n} |y_i - (\mathbf{w} \cdot \mathbf{x}_i + b)|\)
- Penalty function: \(\lambda |\mathbf{w}|^p\)
- Resulting algorithm: Ridge regression

### Comparing \(\epsilon\)-SVR with popular regression methods

<table>
<thead>
<tr>
<th>Loss function</th>
<th>Penalty function</th>
<th>Resulting algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\epsilon)-insensitive loss: (\sum_{i=1}^{n} \max(0,</td>
<td>y_i - (\mathbf{w} \cdot \mathbf{x}_i + b) - \epsilon)</td>
<td>)</td>
</tr>
<tr>
<td>Quadratic (\epsilon)-insensitive loss: (\sum_{i=1}^{n} \max(0,</td>
<td>y_i - (\mathbf{w} \cdot \Phi(\mathbf{c}_i) + b) - \epsilon)</td>
<td>^2)</td>
</tr>
<tr>
<td>Quadratic (\epsilon)-insensitive loss: (\sum_{i=1}^{n} \max(0,</td>
<td>y_i - (\mathbf{w} \cdot \Phi(\mathbf{c}_i) + b) - \epsilon)</td>
<td>^2)</td>
</tr>
<tr>
<td>Squared error: (\sum_{i=1}^{n} (y_i - (\mathbf{w} \cdot \mathbf{x}_i + b))^2)</td>
<td>(-)</td>
<td>Multiple Linear Regression (least squares formulation)</td>
</tr>
<tr>
<td>Squared error: (\sum_{i=1}^{n} (y_i - (\mathbf{w} \cdot \mathbf{x}_i + b))^2)</td>
<td>(\lambda</td>
<td>\mathbf{w}</td>
</tr>
<tr>
<td>Linear error: (\sum_{i=1}^{n}</td>
<td>y_i - (\mathbf{w} \cdot \mathbf{x}_i + b)</td>
<td>)</td>
</tr>
<tr>
<td>Squared error: (\sum_{i=1}^{n}</td>
<td>y_i - (\mathbf{w} \cdot \mathbf{x}_i + b)</td>
<td>^2)</td>
</tr>
<tr>
<td>Squared error: (\sum_{i=1}^{n}</td>
<td>y_i - (\mathbf{w} \cdot \mathbf{x}_i + b)</td>
<td>^2)</td>
</tr>
</tbody>
</table>
Applying $\varepsilon$-SVR to real data

In the absence of domain knowledge about decision functions, it is recommended to optimize the following parameters (e.g., by cross-validation using grid-search):

- parameter $C$
- parameter $\varepsilon$
- kernel parameters (e.g., degree of polynomial)

Notice that parameter $\varepsilon$ depends on the ranges of variables in the dataset; therefore it is recommended to normalize/re-scale data prior to applying $\varepsilon$-SVR.

Novelty detection with SVM-based methods
What is it about?

• Find the simplest and most compact region in the space of predictors where the majority of data samples “live” (i.e., with the highest density of samples).
• Build a decision function that takes value +1 in this region and -1 elsewhere.
• Once we have such a decision function, we can identify novel or outlier samples/patients in the data.

Key assumptions

• We do not know classes/labels of samples (positive or negative) in the data available for learning ➔ this is not a classification problem
• All positive samples are similar but each negative sample can be different in its own way

Thus, do not need to collect data for negative samples!
Sample applications

Discover deviations in sample handling protocol when doing quality control of assays.

Protein Y

- Samples with low quality of processing from the lab of Dr. Smith
- Samples with high-quality of processing
- Samples with low quality of processing from infants

Protein X

- Samples with low quality of processing from ICU patients
- Samples with low quality of processing from patients with lung cancer

Sample applications

Identify websites that discuss benefits of proven cancer treatments.

On Presentation of one-class SVM...

When we previously discussed SVM classification and regression, we first introduced the version of the method for linear data, then discussed the non-linear version. We will follow a similar course in the discussion of one-class SVMs but with one fundamental difference. Unlike SVMs and SVR, only non-linear one-class SVM is designed to be applied and makes practical sense. Furthermore, one-class SVM should be used with special types of kernel functions. So, even though we will describe linear one-class SVM, it is presented here only for educational purposes and its practical meaning may be limited.
Linear one-class SVM

**Main idea:** Find the maximal gap hyperplane that separates data from the origin (i.e., the only member of the second class is the origin).

The equation is:

\[ \vec{w} \cdot \vec{x} + b = +1 \]

\[ \vec{w} \cdot \vec{x} + b = 0 \]

\[ \vec{w} \cdot \vec{x} + b = -1 \]

 Origin is the only member of the second class

Gene X

Gene Y

One-class SVM seeks the most compact region where the majority of data points live; so we are interested in another hyperplane:

\[ \vec{w} \cdot \vec{x} + b = 0 \]

Origin is the only member of the second class

Gene X

Gene Y

Star Negative object (y=-1)  Red Positive objects (y=+1)
Formulation of one-class SVM: linear case

Given training data: \( \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N \in \mathbb{R}^n \)

Find \( f(\vec{x}) = \text{sign}(\vec{w} \cdot \vec{x} + b) \)

by minimizing \( \frac{1}{2} \|\vec{w}\|^2 + \frac{1}{vN} \sum_{i=1}^{N} \xi_i + b \)

subject to constraints:

\[
\vec{w} \cdot \vec{x} + b \geq -\xi_i \\
\xi_i \geq 0 \\
\text{for } i = 1, \ldots, N.
\]

i.e., the decision function should be positive in all training samples except for small deviations.

Formulation of one-class SVM: linear and non-linear cases

**Linear case**

Find \( f(\vec{x}) = \text{sign}(\vec{w} \cdot \vec{x} + b) \)

by minimizing \( \frac{1}{2} \|\vec{w}\|^2 + \frac{1}{vN} \sum_{i=1}^{N} \xi_i + b \)

subject to constraints:

\[
\vec{w} \cdot \vec{x} + b \geq -\xi_i \\
\xi_i \geq 0 \\
\text{for } i = 1, \ldots, N.
\]

**Non-linear case**

(use “kernel trick”)

Find \( f(\vec{x}) = \text{sign}(\vec{w} \cdot \Phi(\vec{x}) + b) \)

by minimizing \( \frac{1}{2} \|\vec{w}\|^2 + \frac{1}{vN} \sum_{i=1}^{N} \xi_i + b \)

subject to constraints:

\[
\vec{w} \cdot \Phi(\vec{x}) + b \geq -\xi_i \\
\xi_i \geq 0 \\
\text{for } i = 1, \ldots, N.
\]
Non-linear one-class SVM

We want to find the simplest and most compact region enclosing the majority of data points.

Use a kernel to map all points to a sphere with unit radius (e.g., use Gaussian kernel)

\[ \Phi(x_i) \]

Feature space

Non-linear one-class SVM

\[ \tilde{w} \cdot \Phi(x) + b = 0 \]

Feature space

Solve linear one-class SVM problem in the feature space
Non-linear one-class SVM

Consider the corresponding decision surface (image) in the input space – it is exactly what we were looking for!

More about one-class SVM

• One-class SVMs inherit most of properties of SVMs for binary classification (e.g., “kernel trick”, sample efficiency, ease of finding of a solution by efficient optimization method, etc.);
• The choice of parameter $\nu$ significantly affects the resulting decision surface.
• The choice of origin is arbitrary and also significantly affects the decision surface returned by the algorithm.
More about one-class SVM

Support vector clustering

Contributed by Nikita Lytkin
Goal of clustering (aka class discovery)

Given a heterogeneous set of data points $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N \in \mathbb{R}^n$
Assign labels $y_1, y_2, \ldots, y_N \in \{1, 2, \ldots, K\}$ such that points with the same label are highly “similar” to each other and are distinctly different from the rest

Support vector domain description

- Support Vector Domain Description (SVDD) of the data is a set of vectors lying on the surface of the smallest hyper-sphere enclosing all data points in a feature space
  - These surface points, called Support Vectors, lie in low density regions in the input space and are good indicators of cluster boundaries
Outline of Support Vector Clustering

- Data points that are support vectors of the minimal enclosing hyper-sphere are taken to be the cluster boundary points.
- All other points are assigned to clusters using the boundary points.

Cluster assignment in SVC

- Two points $x_i$ and $x_j$ belong to the same cluster (i.e., have the same label) if every point of the line segment $(x_i, x_j)$ projected into the feature space lies within the hyper-sphere.
Cluster assignment in SVC (continued)

• Point-wise adjacency matrix is constructed by testing the line segments between every pair of points
• Connected components are extracted
• Points belonging to the same connected component are assigned to the same cluster

SVM-based variable selection
Understanding the weight vector $w$

Recall standard SVM formulation:

Find $w$ and $b$ that minimize

$$\frac{1}{2} \|w\|^2 \text{ subject to } y_i (\langle w, x_i \rangle + b) \geq 1$$

for $i = 1, \ldots, N$.

Use classifier: $f(\tilde{x}) = \text{sign}(\langle w, \tilde{x} \rangle + b)$

- The weight vector $\tilde{w}$ contains as many elements as there are input variables in the dataset, i.e. $\tilde{w} \in \mathbb{R}^n$.
- The magnitude of each element denotes importance of the corresponding variable for classification task.

**Understanding the weight vector $w$**

- **$\tilde{w} = (1,1)$**
  - $1x_1 + 1x_2 + b = 0$
  - $X_1$ and $X_2$ are equally important

- **$\tilde{w} = (1,0)$**
  - $1x_1 + 0x_2 + b = 0$
  - $X_1$ is important, $X_2$ is not

- **$\tilde{w} = (0,1)$**
  - $0x_1 + 1x_2 + b = 0$
  - $X_2$ is important, $X_1$ is not

- **$\tilde{w} = (1,1,0)$**
  - $1x_1 + 1x_2 + 0x_3 + b = 0$
  - $X_1$ and $X_2$ are equally important, $X_3$ is not
Understanding the weight vector \( \mathbf{w} \)

- In the true model, \( X_1 \) is causal and \( X_2 \) is redundant.
- SVM decision surface implies that \( X_1 \) and \( X_2 \) are equally important; thus it is locally causally inconsistent.
- There exists a causally consistent decision surface for this example.
- Causal discovery algorithms can identify that \( X_1 \) is causal and \( X_2 \) is redundant.

Simple SVM-based variable selection algorithm

**Algorithm:**

1. Train SVM classifier using data for all variables to estimate vector \( \mathbf{w} \).
2. Rank each variable based on the magnitude of the corresponding element in vector \( \mathbf{w} \).
3. Using the above ranking of variables, select the smallest nested subset of variables that achieves the best SVM prediction accuracy.
Simple SVM-based variable selection algorithm

Consider that we have 7 variables: \(X_1, X_2, X_3, X_4, X_5, X_6, X_7\)
The vector \(\vec{w}\) is: \((0.1, 0.3, 0.4, 0.01, 0.9, -0.99, 0.2)\)
The ranking of variables is: \(X_6, X_5, X_3, X_2, X_7, X_1, X_4\)

<table>
<thead>
<tr>
<th>Subset of variables</th>
<th>Classification accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X_6)</td>
<td>0.920</td>
</tr>
<tr>
<td>(X_6)</td>
<td>0.920</td>
</tr>
<tr>
<td>(X_6)</td>
<td>0.919</td>
</tr>
<tr>
<td>(X_6)</td>
<td>0.852</td>
</tr>
<tr>
<td>(X_6)</td>
<td>0.843</td>
</tr>
<tr>
<td>(X_6)</td>
<td>0.832</td>
</tr>
<tr>
<td>(X_6)</td>
<td>0.821</td>
</tr>
</tbody>
</table>

Select the following variable subset: \(X_6, X_5, X_3, X_2, X_7\)

Simple SVM-based variable selection algorithm

- SVM weights are not locally causally consistent \(\Rightarrow\) we may end up with a variable subset that is not causal and not necessarily the most compact one.
- The magnitude of a variable in vector \(\vec{w}\) estimates the effect of removing that variable on the objective function of SVM (e.g., function that we want to minimize). However, this algorithm becomes suboptimal when considering effect of removing several variables at a time... This pitfall is addressed in the SVM-RFE algorithm that is presented next.
SVM-RFE variable selection algorithm

Algorithm:
1. Initialize \( V \) to all variables in the data
2. Repeat
3. Train SVM classifier using data for variables in \( V \) to estimate vector \( \vec{w} \)
4. Estimate prediction accuracy of variables in \( V \) using the above SVM classifier (e.g., by cross-validation)
5. Remove from \( V \) a variable (or a subset of variables) with the smallest magnitude of the corresponding element in vector \( \vec{w} \)
6. Until there are no variables in \( V \)
7. Select the smallest subset of variables with the best prediction accuracy

• Unlike simple SVM-based variable selection algorithm, SVM-RFE estimates vector \( \vec{w} \) many times to establish ranking of the variables.
• Notice that the prediction accuracy should be estimated at each step in an unbiased fashion, e.g. by cross-validation.
SVM variable selection in feature space

The real power of SVMs comes with application of the kernel trick that maps data to a much higher dimensional space ("feature space") where the data is linearly separable.

![Image](image.png)

SVM variable selection in feature space

- We have data for 100 SNPs \((X_1, \ldots, X_{100})\) and some phenotype.
- We allow up to 3\(^{rd}\) order interactions, e.g. we consider:
  - \(X_1, \ldots, X_{100}\)
  - \(X_1^2, X_1 X_2, X_1 X_3, \ldots, X_1 X_{100}, \ldots, X_{100}^2\)
  - \(X_1^3, X_1 X_2 X_3, X_1 X_2 X_4, \ldots, X_1 X_{99} X_{100}, \ldots, X_{100}^3\)

- **Task**: find the smallest subset of features (either SNPs or their interactions) that achieves the best predictive accuracy of the phenotype.
- **Challenge**: If we have limited sample, we cannot explicitly construct and evaluate all SNPs and their interactions (176,851 features in total) as it is done in classical statistics.
SVM variable selection in feature space

**Heuristic solution:** Apply algorithm SVM-FSMB that:

1. Uses SVMs with polynomial kernel of degree 3 and selects M features (not necessarily input variables!) that have largest weights in the feature space.
   
   E.g., the algorithm can select features like: $X_{10}$, $(X_1X_2)$, $(X_9X_2X_{22})$, $(X_7^2X_{98})$, and so on.

2. Apply HITON-MB Markov blanket algorithm to find the Markov blanket of the phenotype using M features from step 1.

Computing posterior class probabilities for SVM classifiers
Output of SVM classifier

1. SVMs output a class label (positive or negative) for each sample: \(\text{sign}(\vec{w} \cdot \vec{x} + b)\)

2. One can also compute distance from the hyperplane that separates classes, e.g. \(\vec{w} \cdot \vec{x} + b\)
   These distances can be used to compute performance metrics like area under ROC curve.

**Question:** How can one use SVMs to estimate posterior class probabilities, i.e., \(P(\text{class positive} \mid \text{sample } x)\)?

---

Simple binning method

1. Train SVM classifier in the *Training set*.

2. Apply it to the *Validation set* and compute distances from the hyperplane to each sample.

<table>
<thead>
<tr>
<th>Sample #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>...</th>
<th>98</th>
<th>99</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance</td>
<td>2</td>
<td>-1</td>
<td>8</td>
<td>3</td>
<td>4</td>
<td>...</td>
<td>-2</td>
<td>0.3</td>
<td>0.8</td>
</tr>
</tbody>
</table>

3. Create a histogram with \(Q\) (e.g., say 10) bins using the above distances. Each bin has an upper and lower value in terms of distance.
Simple binning method

4. Given a new sample from the **Testing set**, place it in the corresponding bin.

   E.g., sample #382 has distance to hyperplane = 1, so it is placed in the bin [0, 2.5]

![Histogram](histogram.png)

5. Compute probability \( P(\text{positive class} \mid \text{sample} \#382) \) as a fraction of true positives in this bin.

   E.g., this bin has 22 samples (from the **Validation set**), out of which 17 are positive ones, so we compute \( P(\text{positive class} \mid \text{sample} \#382) = 17/22 = 0.77 \)

---

Platt’s method

Convert distances output by SVM to probabilities by passing them through the sigmoid filter:

\[
P(\text{positive class} \mid \text{sample}) = \frac{1}{1 + \exp(Ad + B)}
\]

where \( d \) is the distance from hyperplane and \( A \) and \( B \) are parameters.
Platt’s method

1. Train SVM classifier in the *Training set*.

2. Apply it to the *Validation set* and compute distances from the hyperplane to each sample.

<table>
<thead>
<tr>
<th>Sample #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>...</th>
<th>98</th>
<th>99</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance</td>
<td>2</td>
<td>-1</td>
<td>8</td>
<td>3</td>
<td>4</td>
<td></td>
<td>-2</td>
<td>0.3</td>
<td>0.8</td>
</tr>
</tbody>
</table>

3. Determine parameters $A$ and $B$ of the sigmoid function by minimizing the negative log likelihood of the data from the *Validation set*.


---

Part 3

- **Case studies** *(taken from our research)*
  1. Classification of cancer gene expression microarray data
  2. Text categorization in biomedicine
  3. Prediction of clinical laboratory values
  4. Modeling clinical judgment
  5. Using SVMs for feature selection
  6. Outlier detection in ovarian cancer proteomics data

- Software
- Conclusions
- Bibliography
1. Classification of cancer gene expression microarray data

Comprehensive evaluation of algorithms for classification of cancer microarray data

**Main goals:**
- Find the best performing decision support algorithms for cancer diagnosis from microarray gene expression data;
- Investigate benefits of using gene selection and ensemble classification methods.
Classifiers

- K-Nearest Neighbors (KNN)
- Backpropagation Neural Networks (NN)
- Probabilistic Neural Networks (PNN)
- Multi-Class SVM: One-Versus-Rest (OVR)
- Multi-Class SVM: One-Versus-One (OVO)
- Multi-Class SVM: DAGSVM
- Multi-Class SVM by Weston & Watkins (WW)
- Multi-Class SVM by Crammer & Singer (CS)
- Weighted Voting: One-Versus-Rest
- Weighted Voting: One-Versus-One
- Decision Trees: CART

Ensemble classifiers
Gene selection methods

1. Signal-to-noise (S2N) ratio in one-versus-rest (OVR) fashion;
2. Signal-to-noise (S2N) ratio in one-versus-one (OVO) fashion;
3. Kruskal-Wallis nonparametric one-way ANOVA (KW);
4. Ratio of genes between categories to within-category sum of squares (BW).

Performance metrics and statistical comparison

1. Accuracy
   + can compare to previous studies
   + easy to interpret & simplifies statistical comparison

2. Relative classifier information (RCI)
   + easy to interpret & simplifies statistical comparison
   + not sensitive to distribution of classes
   + accounts for difficulty of a decision problem

• Randomized permutation testing to compare accuracies of the classifiers ($\alpha=0.05$)
## Microarray Datasets

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>Number of Samples</th>
<th>Variables (genes)</th>
<th>Categories</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>11_Tumors</td>
<td>174</td>
<td>12533</td>
<td>11</td>
<td>Su, 2001</td>
</tr>
<tr>
<td>14_Tumors</td>
<td>308</td>
<td>15009</td>
<td>26</td>
<td>Ramaswamy, 2001</td>
</tr>
<tr>
<td>9_Tumors</td>
<td>60</td>
<td>5726</td>
<td>9</td>
<td>Staunton, 2001</td>
</tr>
<tr>
<td>Brain_Tumor1</td>
<td>90</td>
<td>5920</td>
<td>5</td>
<td>Pomeroy, 2002</td>
</tr>
<tr>
<td>Brain_Tumor2</td>
<td>50</td>
<td>10367</td>
<td>4</td>
<td>Nutt, 2003</td>
</tr>
<tr>
<td>Leukemia1</td>
<td>72</td>
<td>5327</td>
<td>3</td>
<td>Golub, 1999</td>
</tr>
<tr>
<td>Leukemia2</td>
<td>72</td>
<td>11225</td>
<td>3</td>
<td>Armstrong, 2002</td>
</tr>
<tr>
<td>Lung_Cancer</td>
<td>203</td>
<td>12600</td>
<td>5</td>
<td>Bhattacherjee, 2001</td>
</tr>
<tr>
<td>SRBCT</td>
<td>83</td>
<td>2308</td>
<td>4</td>
<td>Khan, 2001</td>
</tr>
<tr>
<td>Prostate_Tumor</td>
<td>102</td>
<td>10509</td>
<td>2</td>
<td>Singh, 2002</td>
</tr>
<tr>
<td>DLBCL</td>
<td>77</td>
<td>5469</td>
<td>2</td>
<td>Shipp, 2002</td>
</tr>
</tbody>
</table>

**Total:**
- ~1300 samples
- 74 diagnostic categories
- 41 cancer types and 12 normal tissue types

---

## Summary of Methods and Datasets

### Cross-Validation Designs (2)
- 10-Fold CV
- LOOCV

### Gene Selection Methods (4)
- S2N One-Versus-Rest
- S2N One-Versus-One
- Non-param. ANOVA
- BW ratio

### Performance Metrics (2)
- Accuracy
- RCI

### Statistical Comparison
- Randomized permutation testing

### Classifiers (11)
- One-Versus-Rest
- One-Versus-One
- DAGSVM
- Method by WW
- Method by CS
- KNN
- Backprop. NN
- Prob. NN
- Decision Trees

### Ensemble Classifiers (7)
- Based on MC-SVM outputs
  - Majority Voting
  - MC-SVM OVR
  - MC-SVM OVO
  - MC-SVM DAGSVM
- Decision Trees
- Based on outputs of all classifiers
  - Majority Voting
  - Decision Trees

### Gene Expression Datasets (11)
- 11_Tumors
- 14_Tumors
- 9_Tumors
- Brain_Tumor1
- Brain_Tumor2
- Leukemia1
- Leukemia2
- Lung_Cancer
- SRBCT
- Prostate_Tumor
- DLBCL
Results without gene selection

Diagram showing accuracy percentages for different datasets and models.

Results with gene selection

Improvement of diagnostic performance by gene selection (averages for the four datasets)

Diagram showing improvement in accuracy with and without gene selection.

Diagnostic performance before and after gene selection

Bar charts comparing accuracy before and after gene selection.

Average reduction of genes is 10-30 times
Comparison with previously published results

Summary of results

- Multi-class SVMs are the best family among the tested algorithms outperforming KNN, NN, PNN, DT, and WV.
- Gene selection in some cases improves classification performance of all classifiers, especially of non-SVM algorithms;
- Ensemble classification does not improve performance of SVM and other classifiers;
- Results obtained by SVMs favorably compare with the literature.
Random Forest (RF) classifiers

- **Appealing properties**
  - Work when \# of predictors > \# of samples
  - Embedded gene selection
  - Incorporate interactions
  - Based on theory of ensemble learning
  - Can work with binary & multiclass tasks
  - Does not require much fine-tuning of parameters

- **Strong theoretical claims**

- **Empirical evidence**: (Diaz-Uriarte and Alvarez de Andres, *BMC Bioinformatics, 2006*) reported superior classification performance of RFs compared to SVMs and other methods

---

**Key principles of RF classifiers**

1. Generate bootstrap samples
2. Random gene selection
3. Fit unpruned decision trees
4. Apply to testing data & combine predictions
Results without gene selection

- SVMs nominally outperform RFs is 15 datasets, RFs outperform SVMs in 4 datasets, algorithms are exactly the same in 3 datasets.
- In 7 datasets SVMs outperform RFs statistically significantly.
- On average, the performance advantage of SVMs is 0.033 AUC and 0.057 RCI.

Results with gene selection

- SVMs nominally outperform RFs is 17 datasets, RFs outperform SVMs in 3 datasets, algorithms are exactly the same in 2 datasets.
- In 1 dataset SVMs outperform RFs statistically significantly.
- On average, the performance advantage of SVMs is 0.028 AUC and 0.047 RCI.
2. Text categorization in biomedicine

Models to categorize content and quality:
Main idea

1. Utilize existing (or easy to build) training corpora

2. Simple document representations (i.e., typically stemmed and weighted words in title and abstract, Mesh terms if available; occasionally addition of Metamap CUIs, author info) as “bag-of-words”
Models to categorize content and quality: Main idea

1. SVM models have excellent ability to identify high-quality PubMed documents according to ACPJ gold standard

2. SVM models have better classification performance than PageRank, Yahoo ranks, Impact Factor, Web Page hit counts, and bibliometric citation counts on the Web according to ACPJ gold standard

3. Train SVM models that capture implicit categories of meaning or quality criteria

4. Evaluate models’ performances - with nested cross-validation or other appropriate error estimators - use primarily AUC as well as other metrics (sensitivity, specificity, PPV, Precision/Recall curves, HIT curves, etc.)

5. Evaluate performance prospectively & compare to prior cross-validation estimates

4. Evaluate models’ performances - with nested cross-validation or other appropriate error estimators - use primarily AUC as well as other metrics (sensitivity, specificity, PPV, Precision/Recall curves, HIT curves, etc.)

5. Evaluate performance prospectively & compare to prior cross-validation estimates
Models to categorize content and quality: Some notable results

<table>
<thead>
<tr>
<th>Gold standard: SSOAB</th>
<th>Area under the ROC curve*</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSOAB-specific filters</td>
<td>0.893</td>
</tr>
<tr>
<td>Citation Count</td>
<td>0.791</td>
</tr>
<tr>
<td>ACPJ Txmt-specific filters</td>
<td>0.548</td>
</tr>
<tr>
<td>Impact Factor (2001)</td>
<td>0.549</td>
</tr>
<tr>
<td>Impact Factor (2005)</td>
<td>0.558</td>
</tr>
</tbody>
</table>

3. SVM models have better classification performance than PageRank, Impact Factor and Citation count in Medline for SSOAB gold standard

4. SVM models have better sensitivity/specificity in PubMed than CQFs at comparable thresholds according to ACPJ gold standard

Other applications of SVMs to text categorization

<table>
<thead>
<tr>
<th>Model</th>
<th>Area Under the Curve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine Learning Models</td>
<td>0.93</td>
</tr>
<tr>
<td>Quackometer*</td>
<td>0.67</td>
</tr>
<tr>
<td>Google</td>
<td>0.63</td>
</tr>
</tbody>
</table>

1. Identifying Web Pages with misleading treatment information according to special purpose gold standard (Quack Watch). SVM models outperform Quackometer and Google ranks in the tested domain of cancer treatment.

2. Prediction of future paper citation and instrumentality of citations
3. Prediction of clinical laboratory values

Dataset generation and experimental design

- StarPanel database contains $\sim 8 \cdot 10^6$ lab measurements of $\sim 100,000$ inpatients from Vanderbilt University Medical Center.
- Lab measurements were taken between 01/1998 and 10/2002.

For each combination of lab test and normal range, we generated the following datasets.

- 01/1998-05/2001: Training
- 06/2001-10/2002: Testing
- Validation (25% of Training)
Query-based approach for prediction of clinical cab values

Classification results

Including cases with $K=0$ (i.e. samples with no prior lab measurements)

<table>
<thead>
<tr>
<th>Range of normal values</th>
<th>&gt;1</th>
<th>&lt;99</th>
<th>[1, 99]</th>
<th>&gt;2.5</th>
<th>&gt;97.5</th>
<th>[2.5, 97.5]</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUN</td>
<td>75.9%</td>
<td>93.4%</td>
<td>88.5%</td>
<td>81.8%</td>
<td>92.2%</td>
<td>66.9%</td>
</tr>
<tr>
<td>Ca</td>
<td>67.5%</td>
<td>80.4%</td>
<td>55.5%</td>
<td>77.4%</td>
<td>70.8%</td>
<td>60.0%</td>
</tr>
<tr>
<td>Calo</td>
<td>63.5%</td>
<td>52.9%</td>
<td>58.8%</td>
<td>46.4%</td>
<td>66.3%</td>
<td>58.7%</td>
</tr>
<tr>
<td>CO2</td>
<td>77.3%</td>
<td>88.0%</td>
<td>53.4%</td>
<td>77.5%</td>
<td>90.5%</td>
<td>58.1%</td>
</tr>
<tr>
<td>Creat</td>
<td>62.2%</td>
<td>88.4%</td>
<td>83.5%</td>
<td>88.4%</td>
<td>94.9%</td>
<td>83.8%</td>
</tr>
<tr>
<td>Mg</td>
<td>58.4%</td>
<td>71.8%</td>
<td>64.2%</td>
<td>67.0%</td>
<td>72.5%</td>
<td>62.1%</td>
</tr>
<tr>
<td>Osmol</td>
<td>77.9%</td>
<td>64.8%</td>
<td>65.2%</td>
<td>79.2%</td>
<td>82.4%</td>
<td>71.5%</td>
</tr>
<tr>
<td>PCV</td>
<td>62.3%</td>
<td>91.6%</td>
<td>69.7%</td>
<td>76.5%</td>
<td>84.6%</td>
<td>70.2%</td>
</tr>
<tr>
<td>Phos</td>
<td>70.8%</td>
<td>75.4%</td>
<td>60.4%</td>
<td>68.0%</td>
<td>81.8%</td>
<td>65.9%</td>
</tr>
</tbody>
</table>

Excluding cases with $K=0$ (i.e. samples with no prior lab measurements)

<table>
<thead>
<tr>
<th>Range of normal values</th>
<th>&gt;1</th>
<th>&lt;99</th>
<th>[1, 99]</th>
<th>&gt;2.5</th>
<th>&gt;97.5</th>
<th>[2.5, 97.5]</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUN</td>
<td>80.4%</td>
<td>99.1%</td>
<td>76.6%</td>
<td>87.1%</td>
<td>98.2%</td>
<td>70.7%</td>
</tr>
<tr>
<td>Ca</td>
<td>72.8%</td>
<td>93.4%</td>
<td>55.6%</td>
<td>81.4%</td>
<td>81.4%</td>
<td>63.4%</td>
</tr>
<tr>
<td>Calo</td>
<td>74.1%</td>
<td>60.0%</td>
<td>50.1%</td>
<td>64.7%</td>
<td>72.3%</td>
<td>57.7%</td>
</tr>
<tr>
<td>CO2</td>
<td>82.0%</td>
<td>93.6%</td>
<td>59.8%</td>
<td>84.4%</td>
<td>94.5%</td>
<td>56.3%</td>
</tr>
<tr>
<td>Creat</td>
<td>62.8%</td>
<td>97.7%</td>
<td>89.1%</td>
<td>91.5%</td>
<td>98.1%</td>
<td>87.7%</td>
</tr>
<tr>
<td>Mg</td>
<td>56.9%</td>
<td>70.0%</td>
<td>49.1%</td>
<td>58.6%</td>
<td>76.9%</td>
<td>59.1%</td>
</tr>
<tr>
<td>Osmol</td>
<td>50.9%</td>
<td>60.8%</td>
<td>60.8%</td>
<td>91.0%</td>
<td>90.5%</td>
<td>68.0%</td>
</tr>
<tr>
<td>PCV</td>
<td>74.9%</td>
<td>99.2%</td>
<td>66.3%</td>
<td>80.9%</td>
<td>80.6%</td>
<td>67.1%</td>
</tr>
<tr>
<td>Phos</td>
<td>74.5%</td>
<td>93.6%</td>
<td>64.4%</td>
<td>71.7%</td>
<td>92.2%</td>
<td>69.7%</td>
</tr>
</tbody>
</table>

A total of 84,240 SVM classifiers were built for 16,848 possible data models.
Improving predictive power and parsimony of a BUN model using feature selection

Model description

<table>
<thead>
<tr>
<th>Test name</th>
<th>BUN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of normal values</td>
<td>&lt; 99 perc.</td>
</tr>
<tr>
<td>Data modeling</td>
<td>SRT</td>
</tr>
<tr>
<td>Number of previous measurements</td>
<td>5</td>
</tr>
<tr>
<td>Use variables corresponding to hospitalization units?</td>
<td>Yes</td>
</tr>
<tr>
<td>Number of prior hospitalizations used</td>
<td>2</td>
</tr>
</tbody>
</table>

Dataset description

<table>
<thead>
<tr>
<th>Training set</th>
<th>Validation set</th>
<th>Testing set</th>
</tr>
</thead>
<tbody>
<tr>
<td>N samples (total)</td>
<td>3740</td>
<td>1251</td>
</tr>
<tr>
<td>N abnormal samples</td>
<td>78</td>
<td>27</td>
</tr>
<tr>
<td>N variables</td>
<td>3442</td>
<td>3442</td>
</tr>
</tbody>
</table>

Classification performance (area under ROC curve)

<table>
<thead>
<tr>
<th>All</th>
<th>RFE_Linear</th>
<th>RFE_Poly</th>
<th>HITON_PC</th>
<th>HITON_MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation set</td>
<td>95.29%</td>
<td>98.78%</td>
<td>98.76%</td>
<td>99.12%</td>
</tr>
<tr>
<td>Testing set</td>
<td>94.72%</td>
<td>99.66%</td>
<td>99.63%</td>
<td>99.16%</td>
</tr>
<tr>
<td>Number of features</td>
<td>3442</td>
<td>26</td>
<td>3</td>
<td>11</td>
</tr>
</tbody>
</table>

Features

1. LAB: PM_1(BUN)
2. LAB: PM_2(Cl)
3. LAB: DT(PM_3(K))
4. LAB: DT(PM_3(Creat))
5. LAB: Test Unit J018 (Test Ca, PM 3)
6. LAB: DT(PM_2(Creat))
7. LAB: PM_1(Mg)
8. LAB: PM_3(Mg)
9. LAB: PM_1(PCV)
10. LAB: DM: Gender
11. LAB: Test Unit V HR (Test CaIo, PM 1)
12. LAB: PM_3(Mg)
13. LAB: DT(PM_5(Mg))
14. LAB: PM_1(PCV)
15. LAB: PM_2(BUN)
16. LAB: Test Unit 11NM (Test PCV, PM 2)
17. LAB: Test Unit 7SCC (Test Ca, PM 1)
18. LAB: DT(PM_2(Phos))
19. LAB: DT(PM_3(Creat))
20. LAB: DT(PM_2(Gluc))
21. LAB: DT(PM_5(Creat))
22. DEMO: Hospitalization Unit TVOS
23. LAB: PM_1(Phos)
24. LAB: PM_2(Phos)
25. LAB: Test Unit 11NM (Test K, PM 5)
26. LAB: Test Unit VHR (Test CaIo, PM 1)
4. Modeling clinical judgment

Methodological framework and study outline
Clinical context of experiment

Malignant melanoma is the most dangerous form of skin cancer. Incidence & mortality have been constantly increasing in the last decades.

Physicians and patients

Patients → N=177
76 melanomas - 101 nevi

Dermatologists → N = 6
3 experts - 3 non-experts

Data collection:
Patients seen prospectively, from 1999 to 2002 at Department of Dermatology, S.Chiara Hospital, Trento, Italy. Inclusion criteria: histological diagnosis and >1 digital image available. Diagnoses made in 2004.

<table>
<thead>
<tr>
<th>Features</th>
<th>Lesion location</th>
<th>Family history of melanoma</th>
<th>Irregular Border</th>
<th>Max-diameter</th>
<th>Fitzpatrick’s Photo-type</th>
<th>Number of colors</th>
<th>Min-diameter</th>
<th>Sunburn</th>
<th>Atypical pigmented network</th>
<th>Evolution</th>
<th>Ephelis</th>
<th>Abrupt network cut-off</th>
<th>Age</th>
<th>Lentigos</th>
<th>Regression-Erythema</th>
<th>Age</th>
<th>Comedo-like openings, milia-like cysts</th>
<th>Gender</th>
<th>Asymmetry</th>
<th>Hypo-pigmentation</th>
<th>Telangiectasia</th>
</tr>
</thead>
</table>
Method to explain physician-specific SVM models

Results: Predicting physicians’ judgments

<table>
<thead>
<tr>
<th>Physicians</th>
<th>All (features)</th>
<th>HITON_PC (features)</th>
<th>HITON_MB (features)</th>
<th>RFE (features)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert 1</td>
<td>0.94 (24)</td>
<td>0.92 (4)</td>
<td>0.92 (5)</td>
<td>0.95 (14)</td>
</tr>
<tr>
<td>Expert 2</td>
<td>0.92 (24)</td>
<td>0.89 (7)</td>
<td>0.90 (7)</td>
<td>0.90 (12)</td>
</tr>
<tr>
<td>Expert 3</td>
<td>0.98 (24)</td>
<td>0.95 (4)</td>
<td>0.95 (4)</td>
<td>0.97 (19)</td>
</tr>
<tr>
<td>NonExpert 1</td>
<td>0.92 (24)</td>
<td>0.89 (5)</td>
<td>0.89 (6)</td>
<td>0.90 (22)</td>
</tr>
<tr>
<td>NonExpert 2</td>
<td>1.00 (24)</td>
<td>0.99 (6)</td>
<td>0.99 (6)</td>
<td>0.98 (11)</td>
</tr>
<tr>
<td>NonExpert 3</td>
<td>0.89 (24)</td>
<td>0.89 (4)</td>
<td>0.89 (6)</td>
<td>0.87 (10)</td>
</tr>
</tbody>
</table>
Results: Physician-specific models

Results: Explaining physician agreement

<table>
<thead>
<tr>
<th>Patient 001</th>
<th>Blue veil</th>
<th>irregular border</th>
<th>streaks</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

Expert 1
AUC=0.92
R^2=99%

Expert 3
AUC=0.95
R^2=99%
Results: Explain physician disagreement

<table>
<thead>
<tr>
<th></th>
<th>Blue veil</th>
<th>irregular border</th>
<th>streaks</th>
<th>number of colors</th>
<th>evolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient 002</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>3</td>
<td>no</td>
</tr>
</tbody>
</table>

Results: Guideline compliance

<table>
<thead>
<tr>
<th>Physician</th>
<th>Reported guidelines</th>
<th>Compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experts 1, 2, 3, non-expert 1</td>
<td>Pattern analysis</td>
<td><strong>Non-compliant</strong>: they ignore the majority of features (17 to 20) recommended by pattern analysis.</td>
</tr>
<tr>
<td>Non expert 2</td>
<td>ABCDE rule</td>
<td><strong>Non compliant</strong>: asymmetry, irregular border and evolution are ignored.</td>
</tr>
<tr>
<td>Non expert 3</td>
<td>Non-standard. Reports using 7 features</td>
<td><strong>Non compliant</strong>: 2 out of 7 reported features are ignored while some non-reported ones are not</td>
</tr>
</tbody>
</table>

**On the contrary**: In all guidelines, the more predictors present, the higher the likelihood of melanoma. All physicians were **compliant** with this principle.
5. Using SVMs for feature selection

Feature selection methods

Feature selection methods (non-causal)
- SVM-RFE
- Univariate + wrapper
- Random forest-based
- LARS-Elastic Net
- RELIEF + wrapper
- L0-norm
- Forward stepwise feature selection
- No feature selection

Causal feature selection methods
- HITON-PC
- HITON-MB
- IAMB
- BLCD
- K2MB

This method outputs a Markov blanket of the response variable (under assumptions)
13 real datasets were used to evaluate feature selection methods

<table>
<thead>
<tr>
<th>Dataset name</th>
<th>Domain</th>
<th>Number of variables</th>
<th>Number of samples</th>
<th>Target</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infant_Mortality</td>
<td>Clinical</td>
<td>86</td>
<td>5,337</td>
<td>Died within the first year</td>
<td>discrete</td>
</tr>
<tr>
<td>Ohsumed</td>
<td>Text</td>
<td>14,373</td>
<td>5,000</td>
<td>Relevant to neonatal diseases</td>
<td>continuous</td>
</tr>
<tr>
<td>ACPJ_Etiology</td>
<td>Text</td>
<td>28,228</td>
<td>15,779</td>
<td>Relevant to etiology</td>
<td>continuous</td>
</tr>
<tr>
<td>Lymphoma</td>
<td>Gene expression</td>
<td>7,399</td>
<td>227</td>
<td>3-year survival: dead vs. alive</td>
<td>continuous</td>
</tr>
<tr>
<td>Gisette</td>
<td>Digit recognition</td>
<td>5,000</td>
<td>7,000</td>
<td>Separate 4 from 9</td>
<td>continuous</td>
</tr>
<tr>
<td>Dexter</td>
<td>Text</td>
<td>19,999</td>
<td>600</td>
<td>Relevant to corporate acquisitions</td>
<td>continuous</td>
</tr>
<tr>
<td>Sylva</td>
<td>Ecology</td>
<td>216</td>
<td>14,394</td>
<td>Ponderosa pine vs. everything else</td>
<td>continuous &amp; discrete</td>
</tr>
<tr>
<td>Ovarian_Cancer</td>
<td>Proteomics</td>
<td>2,190</td>
<td>216</td>
<td>Cancer vs. normals</td>
<td>continuous</td>
</tr>
<tr>
<td>Thrombin</td>
<td>Drug discovery</td>
<td>139,351</td>
<td>2,543</td>
<td>Binding to thromin</td>
<td>discrete (binary)</td>
</tr>
<tr>
<td>Breast_Cancer</td>
<td>Gene expression</td>
<td>17,816</td>
<td>286</td>
<td>Estrogen-receptor positive (ER+) vs. ER-</td>
<td>continuous</td>
</tr>
<tr>
<td>Hiva</td>
<td>Drug discovery</td>
<td>1,617</td>
<td>4,229</td>
<td>Activity to AIDS HIV infection</td>
<td>discrete (binary)</td>
</tr>
<tr>
<td>Nova</td>
<td>Text</td>
<td>16,969</td>
<td>1,929</td>
<td>Separate politics from religion topics</td>
<td>discrete (binary)</td>
</tr>
<tr>
<td>Bankruptcy</td>
<td>Financial</td>
<td>147</td>
<td>7,063</td>
<td>Personal bankruptcy</td>
<td>continuous &amp; discrete</td>
</tr>
</tbody>
</table>

Classification performance vs. proportion of selected features

![Classification performance graph](chart.png)
Statistical comparison of predictivity and reduction of features

<table>
<thead>
<tr>
<th></th>
<th>Predictivity</th>
<th></th>
<th>Reduction</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P-value</td>
<td>Nominal winner</td>
<td>P-value</td>
<td>Nominal winner</td>
</tr>
<tr>
<td>SVM-RFE (4 variants)</td>
<td>0.9754</td>
<td>SVM-RFE</td>
<td>0.0046</td>
<td>HITON-PC</td>
</tr>
<tr>
<td></td>
<td>0.8030</td>
<td>SVM-RFE</td>
<td>0.0042</td>
<td>HITON-PC</td>
</tr>
<tr>
<td></td>
<td>0.1312</td>
<td>HITON-PC</td>
<td>0.3634</td>
<td>HITON-PC</td>
</tr>
<tr>
<td></td>
<td>0.1008</td>
<td>HITON-PC</td>
<td>0.6816</td>
<td>SVM-RFE</td>
</tr>
</tbody>
</table>

- Null hypothesis: SVM-RFE and HITON-PC perform the same;
- Use permutation-based statistical test with alpha = 0.05.

Simulated datasets with known causal structure used to compare algorithms

<table>
<thead>
<tr>
<th>Bayesian network</th>
<th>Number of variables</th>
<th>Training samples</th>
<th>Number of selected targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Child10</td>
<td>200</td>
<td>5 x 200, 5 x 500, 1 x 5000</td>
<td>10</td>
</tr>
<tr>
<td>Insurance10</td>
<td>270</td>
<td>5 x 200, 5 x 500, 1 x 5000</td>
<td>10</td>
</tr>
<tr>
<td>Alarm10</td>
<td>370</td>
<td>5 x 200, 5 x 500, 1 x 5000</td>
<td>10</td>
</tr>
<tr>
<td>Hailfinder10</td>
<td>560</td>
<td>5 x 200, 5 x 500, 1 x 5000</td>
<td>10</td>
</tr>
<tr>
<td>Munin</td>
<td>189</td>
<td>5 x 500, 1 x 5000</td>
<td>6</td>
</tr>
<tr>
<td>Pigs</td>
<td>441</td>
<td>5 x 200, 5 x 500, 1 x 5000</td>
<td>10</td>
</tr>
<tr>
<td>Link</td>
<td>724</td>
<td>5 x 200, 5 x 500, 1 x 5000</td>
<td>10</td>
</tr>
<tr>
<td>Lung_Cancer</td>
<td>800</td>
<td>5 x 200, 5 x 500, 1 x 5000</td>
<td>11</td>
</tr>
<tr>
<td>Gene</td>
<td>801</td>
<td>5 x 200, 5 x 500, 1 x 5000</td>
<td>11</td>
</tr>
</tbody>
</table>
Comparison of SVM-RFE and HITON-PC

Comparison of all methods in terms of causal graph distance
Summary results

Causal graph distance from the target

Predictivity

(a) Sample size = 200
(b) Sample size = 500

Statistical comparison of graph distance

<table>
<thead>
<tr>
<th>Sample size = 200</th>
<th>Sample size = 500</th>
<th>Sample size = 5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparison</td>
<td>P-value</td>
<td>Nominal winner</td>
</tr>
<tr>
<td>average HITON-PC-FDR with G^2 test vs. average SVM-RFE</td>
<td>&lt;0.0001</td>
<td>HITON-PC-FDR</td>
</tr>
</tbody>
</table>

- Null hypothesis: SVM-RFE and HITON-PC-FDR perform the same;
- Use permutation-based statistical test with alpha = 0.05.
6. Outlier detection in ovarian cancer proteomics data

Ovarian cancer data

Same set of 216 patients, obtained using the Ciphergen H4 ProteinChip array (dataset 1) and using the Ciphergen WCX2 ProteinChip array (dataset 2).

The gross break at the “benign disease” juncture in dataset 1 and the similarity of the profiles to those in dataset 2 suggest change of protocol in the middle of the first experiment.
Experiments with one-class SVM

Assume that sets \{A, B\} are normal and \{C, D, E, F\} are outliers. Also, assume that we do not know what are normal and outlier samples.

• **Experiment 1**: Train one-class SVM on \{A, B, C\} and test on \{A, B, C\}: Area under ROC curve = \textbf{0.98}

• **Experiment 2**: Train one-class SVM on \{A, C\} and test on \{B, D, E, F\}: Area under ROC curve = \textbf{0.98}
Interactive media and animations

SVM Applets
- http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- http://svm.dcs.rhbnc.ac.uk/pagesnew/GPat.shtml
- http://www.smartlab.dibe.unige.it/Files/sw/Applet%20SVM/svmapplet.html
- http://www.eee.metu.edu.tr/~alatan/Courses/Demo/AppletSVM.html
- http://www.dsl-lab.org/svm_tutorial/demo.html (requires Java 3D)

Animations
- Support Vector Machines:
  - http://www.youtube.com/watch?v=3liCbRZPrZA
  - http://www.youtube.com/watch?v=AC7afKlgGTs
- Support Vector Regression:
  - http://www.cs.ust.hk/irproj/Regularization%20Path/movie/ga0.5lam1.avi

Several SVM implementations for beginners

- GEMS: http://www.gems-system.org
- Weka: http://www.cs.waikato.ac.nz/ml/weka/
- Spider (for Matlab): http://www.kyb.mpg.de/bs/people/spider/
- CLOP (for Matlab): http://clopinet.com/CLOP/
Several SVM implementations for intermediate users

• **LibSVM**: [http://www.csie.ntu.edu.tw/~cjlin/libsvm/](http://www.csie.ntu.edu.tw/~cjlin/libsvm/)
  - General purpose
  - Implements binary SVM, multiclass SVM, SVR, one-class SVM
  - Command-line interface
  - Code/interface for C/C++/C#, Java, Matlab, R, Python, Pearl

• **SVMLight**: [http://svmlight.joachims.org/](http://svmlight.joachims.org/)
  - General purpose (designed for text categorization)
  - Implements binary SVM, multiclass SVM, SVR
  - Command-line interface
  - Code/interface for C/C++, Java, Matlab, Python, Pearl

• Excellent empirical performance for many biomedical datasets.
• Can learn efficiently both simple linear and very complex nonlinear functions by using the “kernel trick”.
• Address outliers and noise by using “slack variables”.
• Internal capacity control (regularization) and overfitting avoidance.
• Do not require direct access to data and can work with only dot-products of data points.
• Require solution of a convex QP optimization problem that has a global minimum and can be solved efficiently. Thus optimizing the SVM model parameters is not subject to heuristic optimization failures that plague other machine learning methods (e.g., neural networks, decision trees).
• Produce sparse models that are defined only by a small subset of training points (“support vectors”).
• Do not have more free parameters than the number of support vectors, irrespective of the number of variables in the dataset.
• Depend on data scaling/normalization.
• Can be used with established data analysis protocols for model selection and error estimation, such as cross-validation.
• When used for feature selection, employ heuristic strategies to identify relevant features and possible mechanisms.

Bibliography
Part 1: Support vector machines for binary classification: classical formulation


Part 1: Basic principles of statistical machine learning

Part 2: Model selection for SVMs


Part 2: SVMs for multicategory data

Part 2: Support vector regression


Part 2: Novelty detection with SVM-based methods and Support Vector Clustering


Part 2: SVM-based variable selection

Part 2: Computing posterior class probabilities for SVM classifiers


Part 3: Classification of cancer gene expression microarray data (Case Study 1)


Part 3: Text Categorization in Biomedicine (Case Study 2)

- Fu L, Aliferis CF: Models for Predicting and Explaining Citation Count of Biomedical Articles. AMIA 2008 Annual Symposium Proceedings 2008.
Part 3: **Modeling clinical judgment**

(Case Study 4)


Part 3: **Using SVMs for feature selection**

(Case Study 5)


Part 3: **Outlier detection in ovarian cancer proteomics data** (Case Study 6)

Thank you for your attention!
Questions/Comments?

Email: Alexander.Statnikov@med.nyu.edu

URL: http://ww.nyuinformatics.org

Upcoming book on SVMs