CSI5180. Machine Learning for Bioinformatics Applications

Support Vector Machines

by

Marcel Turcotte

Version November 6, 2019

Preamble

Preamble

Support Vector Machines

In this lecture, we consider one of the most popular **kernel methods**, the **support vector machines**. We informally discuss their main concepts: **separating hyperplane**, **maximum-margin hyperplane**, the **soft margin**, and **kernel functions**. As a running example, we consider the case of classifying acute leukemias patients using DNA microarray data.

General objective :

- **Explain** in your own words support vector machines

Learning objectives

- **Discuss** the concept of separating hyperplane
- **Explain** why soft margin is needed
- **Describe** in your own words the maximum-margin hyperplane
- **Justify** the need for a kernel function

Reading:

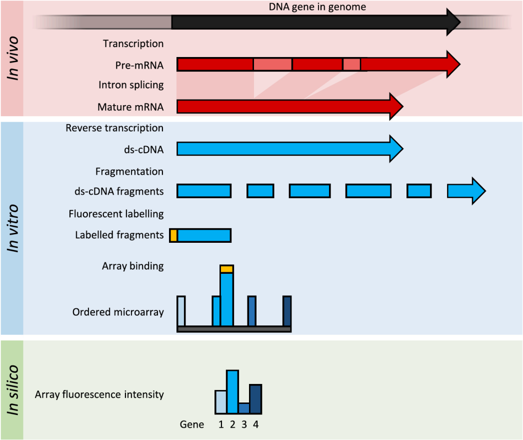
- Noble, W. S. What is a support vector machine? *Nat Biotechnol* **24**:15651567 (2006).

Plan

1. Preamble
2. Problem
3. Introduction
4. Implementation
5. Prologue

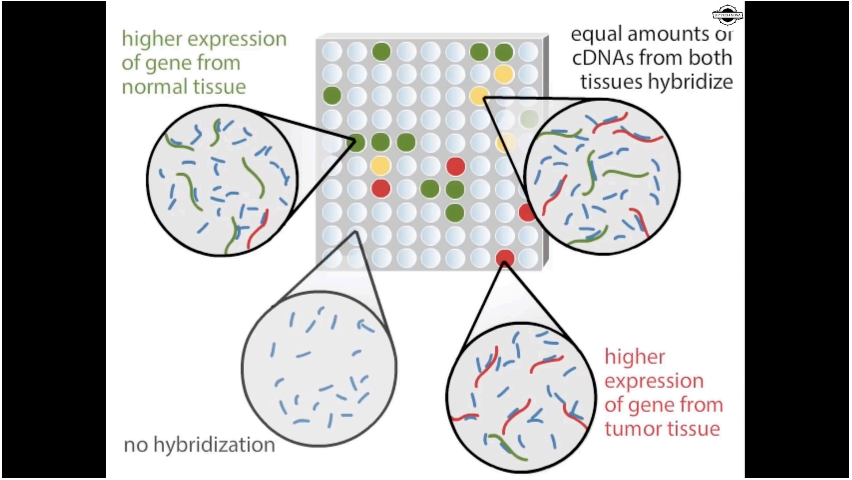
Problem

DNA Microarrays



- Lowe, R., Shirley, N., Bleackley, M., Dolan, S. & Shafee, T. Transcriptomics technologies. *PLoS Comput Biol* **13**, (2017).

DNA microarrays



https://youtu.be/yzBVOCwRanI

Classification of cancer

- Affymetrix microarrays with **6,817 genes**

Classification of cancer

- Affymetrix microarrays with **6,817 genes**
- **38 samples**

Classification of cancer

- Affymetrix microarrays with **6,817 genes**
- **38 samples**
  - **27 samples** from individuals with **acute lymphoblastic leukemia** (**ALL**)

Classification of cancer

- Affymetrix microarrays with **6,817 genes**
- **38 samples**
  - **27 samples** from individuals with **acute lymphoblastic leukemia** (**ALL**)
  - **11 samples** from individuals with **acute myeloid leukemia** (**AML**)

Classification of cancer

- Affymetrix microarrays with **6,817 genes**
- **38 samples**
  - **27 samples** from individuals with **acute lymphoblastic leukemia** (**ALL**)
  - **11 samples** from individuals with **acute myeloid leukemia** (**AML**)

Classification of cancer

- Affymetrix microarrays with **6,817 genes**
- **38 samples**
  - **27 samples** from individuals with **acute lymphoblastic leukemia** (**ALL**)
  - **11 samples** from individuals with **acute myeloid leukemia** (**AML**)

Sources:

- T R Golub, D K Slonim, P Tamayo, C Huard, M Gaasenbeek, J P Mesirov, H Coller, M L Loh, J R Downing, M A Caligiuri, C D Bloomfield, and E S Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science* **286**:5439, 5317, 1999.
- Noble, W. S. What is a support vector machine? *Nat Biotechnol* **24**, 15651567, 2006.

Data/problem

- $\{(x_i, y_i)\}_{i=1}^{N}$

Data/problem

- $\{(x_i, y_i)\}_{i=1}^{N}$
  - $N = 38$

Data/problem

- $\{(x_i, y_i)\}_{i=1}^{N}$
  - $N = 38$
  - Each x_i represents the **expression** of D (6,817) **genes** for given **sample** - a **feature vector** with D dimensions.

Data/problem

- $\{(x_i, y_i)\}_{i=1}^{N}$
  - $N = 38$
  - Each x_i represents the **expression** of D (6,817) **genes** for given **sample** - a **feature vector** with D dimensions.
  - $x_i^{(j)}$ is the value of the **feature** j of the example i, for $j \in 1 \ldots D$ and $i \in 1 \ldots N$. This is the **expression level** of **gene** j for **sample** i.

Data/problem

- $\{(x_i, y_i)\}_{i=1}^{N}$
  - $N = 38$
  - Each x_i represents the **expression** of D (6,817) **genes** for given **sample** - a **feature vector** with D dimensions.
  - $x_i^{(j)}$ is the value of the **feature** j of the example i, for $j \in 1 \ldots D$ and $i \in 1 \ldots N$. This is the **expression level** of **gene** j for **sample** i.
  - The **label** y_i is AML (acute myeloid leukemia) or ALL (acute lymphoblastic leukemia), two classes of acute leukemias.

Data/problem

- $\{(x_i, y_i)\}_{i=1}^{N}$
  - $N = 38$
  - Each x_i represents the **expression** of D (6,817) **genes** for given **sample** - a **feature vector** with D dimensions.
  - $x_i^{(j)}$ is the value of the **feature** j of the example i, for $j \in 1 \dots D$ and $i \in 1 \dots N$. This is the **expression level** of **gene** j for **sample** i.
  - The **label** y_i is AML (acute myeloid leukemia) or ALL (acute lymphoblastic leukemia), two classes of acute leukemias.
    - For reasons that will become evident in the technical part of our discussion, let's use the **label -1 for one of the two classes**, and **1 for the other class**, $y \in \{-1, 1\}$.

Data/problem

- $\{(x_i, y_i)\}_{i=1}^{N}$
  - $N = 38$
  - Each x_i represents the **expression** of D (6,817) **genes** for given **sample** - a **feature vector** with D dimensions.
  - $x_i^{(j)}$ is the value of the **feature** j of the example i, for $j \in 1 \dots D$ and $i \in 1 \dots N$. This is the **expression level** of **gene** j for **sample** i.
  - The **label** y_i is AML (acute myeloid leukemia) or ALL (acute lymphoblastic leukemia), two classes of acute leukemias.
    - For reasons that will become evident in the technical part of our discussion, let's use the **label -1 for one of the two classes**, and **1 for the other class**, $y \in \{-1, 1\}$.
- A **binary classification** task.

Introduction

Introduction

- Just like **hidden Markov models** (**HMM**), **support vector machines** (**SVM**) have a strong theoretical foundation.

Introduction

- Just like **hidden Markov models** (**HMM**), **support vector machines** (**SVM**) have a strong theoretical foundation.
- We **first** introduce the essential concepts **informally**.

Introduction

- Just like **hidden Markov models** (**HMM**), **support vector machines** (**SVM**) have a strong theoretical foundation.
- We **first** introduce the essential concepts **informally**.
  - **Adapted from**:

Introduction

- Just like **hidden Markov models** (**HMM**), **support vector machines** (**SVM**) have a strong theoretical foundation.
- We **first** introduce the essential concepts **informally**.
  - **Adapted from**:
    - Noble, W. S. What is a support vector machine? *Nat Biotechnol* **24**, 15651567, 2006.

Introduction

- Just like **hidden Markov models** (**HMM**), **support vector machines** (**SVM**) have a strong theoretical foundation.
- We **first** introduce the essential concepts **informally**.
  - **Adapted from**:
    - Noble, W. S. What is a support vector machine? *Nat Biotechnol* **24**, 15651567, 2006.
  - We consider two genes, **ZYX** and **MARCKSL1**, so that the data is represented in a two-dimensional plane.

Introduction

- Just like **hidden Markov models** (**HMM**), **support vector machines** (**SVM**) have a strong theoretical foundation.
- We **first** introduce the essential concepts **informally**.
  - **Adapted from**:
    - Noble, W. S. What is a support vector machine? *Nat Biotechnol* **24**, 15651567, 2006.
  - We consider two genes, **ZYX** and **MARCKSL1**, so that the data is represented in a two-dimensional plane.
  - **Four concepts:**

Introduction

- Just like **hidden Markov models** (**HMM**), **support vector machines** (**SVM**) have a strong theoretical foundation.
- We **first** introduce the essential concepts **informally**.
  - **Adapted from**:
    - Noble, W. S. What is a support vector machine? *Nat Biotechnol* **24**, 15651567, 2006.
  - We consider two genes, **ZYX** and **MARCKSL1**, so that the data is represented in a two-dimensional plane.
  - **Four concepts:**
    - **Separating hyperplane**

Introduction

- Just like **hidden Markov models** (**HMM**), **support vector machines** (**SVM**) have a strong theoretical foundation.
- We **first** introduce the essential concepts **informally**.
  - **Adapted from**:
    - Noble, W. S. What is a support vector machine? *Nat Biotechnol* **24**, 15651567, 2006.
  - We consider two genes, **ZYX** and **MARCKSL1**, so that the data is represented in a two-dimensional plane.
  - **Four concepts:**
    - **Separating hyperplane**
    - **Maximum-margin hyperplane**

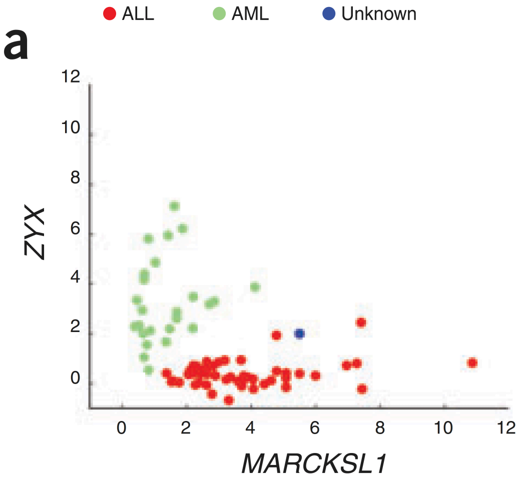
Introduction

- Just like **hidden Markov models** (**HMM**), **support vector machines** (**SVM**) have a strong theoretical foundation.
- We **first** introduce the essential concepts **informally**.
  - **Adapted from**:
    - Noble, W. S. What is a support vector machine? *Nat Biotechnol* **24**, 15651567, 2006.
  - We consider two genes, **ZYX** and **MARCKSL1**, so that the data is represented in a two-dimensional plane.
  - **Four concepts:**
    - **Separating hyperplane**
    - **Maximum-margin hyperplane**
    - **Soft-margin**

Introduction

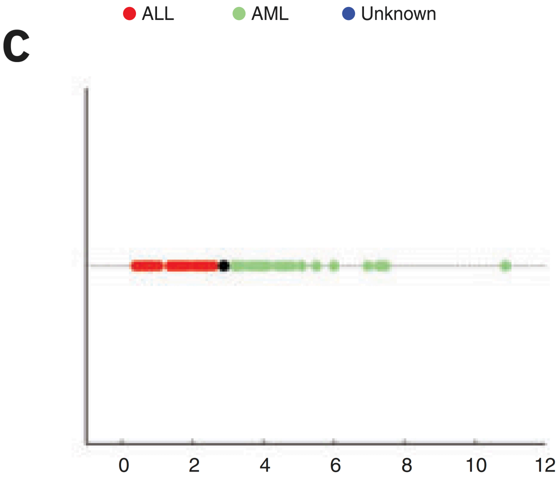
- Just like **hidden Markov models** (**HMM**), **support vector machines** (**SVM**) have a strong theoretical foundation.
- We **first** introduce the essential concepts **informally**.
  - **Adapted from**:
    - Noble, W. S. What is a support vector machine? *Nat Biotechnol* **24**, 15651567, 2006.
  - We consider two genes, **ZYX** and **MARCKSL1**, so that the data is represented in a two-dimensional plane.
  - **Four concepts:**
    - **Separating hyperplane**
    - **Maximum-margin hyperplane**
    - **Soft-margin**
    - **Kernel function**

Separating hyperplane - 2 genes (features)



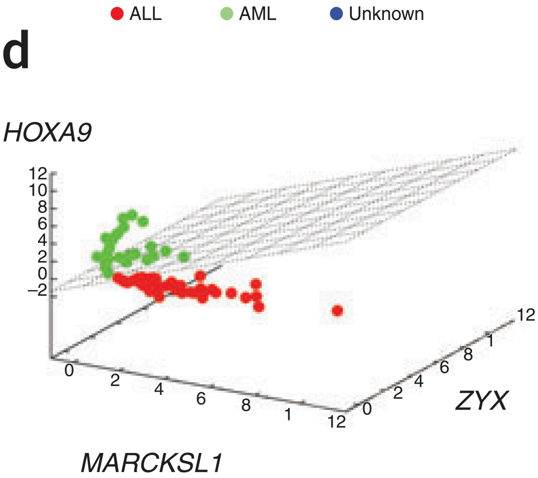
Source: [1] Figure 1a

Separating hyperplane - 1 genes (feature)



Source: [1] Figure 1c

Separating hyperplane - 3 genes (features)

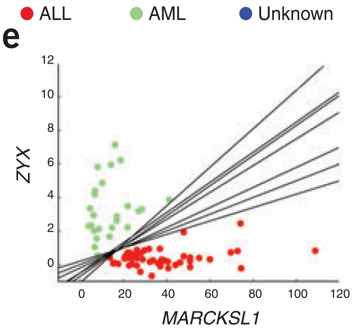


Source: [1] Figure 1d

Separating hyperplane - D genes (features)

- Given D features (genes), $D >> 3$, the **decision boundary** is a **hyperplane**.
- "The general term for a **straight** line in a **high-dimensional space** is a **hyperplane**, and so the **separating hyperplane** is, essentially, the **line that separates the ALL and AML samples**." [1]

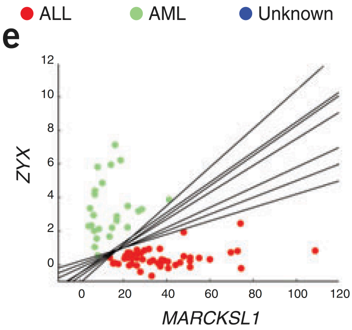
Many separating hyperplanes



Source: [1] Figure 1e

- There are generally infinitely **many separating hyperplanes**, which one to **choose**?

Many separating hyperplanes

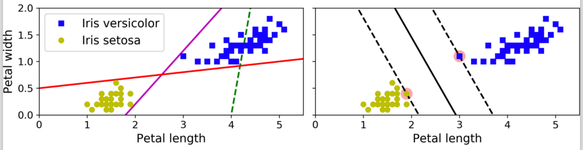


Source: [1] Figure 1e

- There are generally infinitely **many separating hyperplanes**, which one to **choose**?
- What would be a good **guiding principle**?

Maximum-margin hyperplane

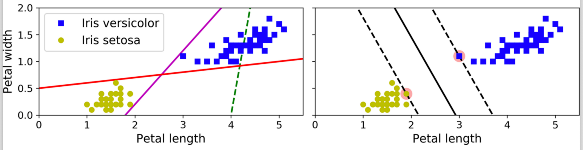
- The **support vectors** are the examples **closest** to the **separating hyperplane**.



Source: [4] Figure 5.1

Maximum-margin hyperplane

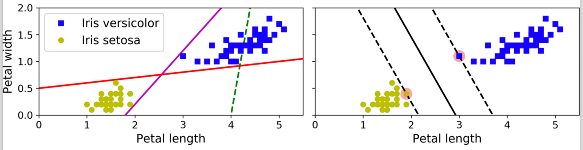
- The **support vectors** are the examples **closest** to the **separating hyperplane**.
- The **margin** is the distance between the **separating hyperplane** (decision boundary) and the **support vectors**.



Source: [4] Figure 5.1

Maximum-margin hyperplane

- The **support vectors** are the examples **closest** to the **separating hyperplane**.
- The **margin** is the distance between the **separating hyperplane** (decision boundary) and the **support vectors**.
- **Problem:** of all possible **separating hyperplanes** find the one with the **largest margin**.



Source: [4] Figure 5.1

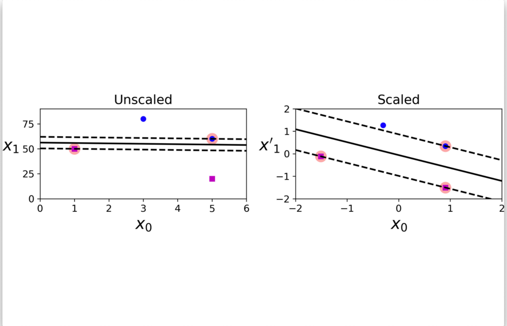
Maximum-margin hyperplane (continued)

- Selecting a **decision boundary** with maximum distance to any example is supported by **statistical learning theory**.

Maximum-margin hyperplane (continued)

- Selecting a **decision boundary** with maximum distance to any example is supported by **statistical learning theory**.
- A **large margin** should decrease the **generalization error** (errors on new cases).

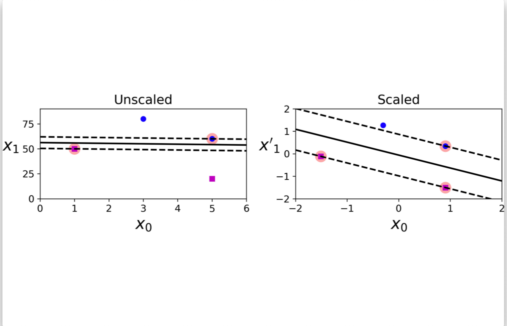
Warning



Source: [4] Figure 5.2

- **Support vector machines** can be negatively affected by **features having different scales**. On the left, x_1 ranges from 0 to 75, whereas x_2 ranges from 0 to 6.

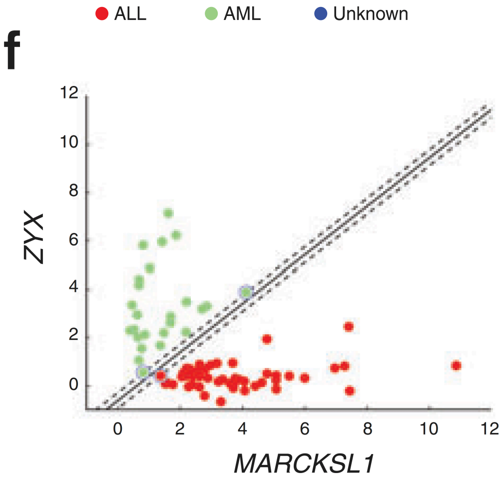
Warning



Source: [4] Figure 5.2

- **Support vector machines** can be negatively affected by **features having different scales**. On the left, x_1 ranges from 0 to 75, whereas x_2 ranges from 0 to 6.
- **Suggestion:** use **sklearn.preprocessing.StandardScaler**.

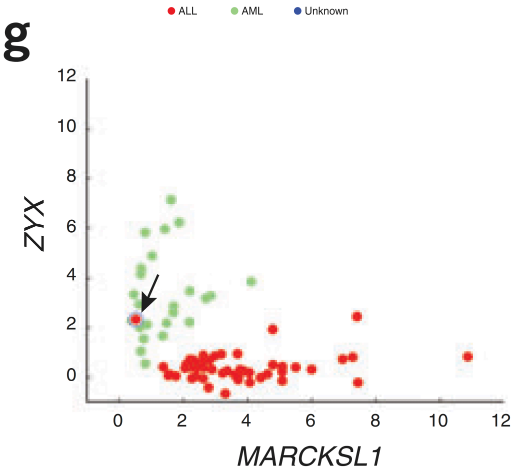
Maximum-margin hyperplane



Source: [1] Figure 1f

Linearly separable?

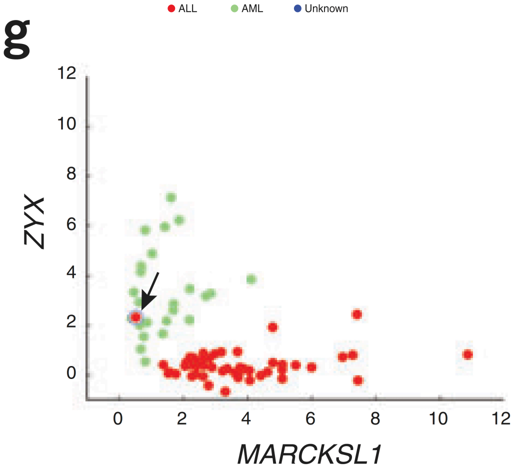
- So far, we have assumed that our data set is **linearly separable**.



Source: [1] Figure 1g

Linearly separable?

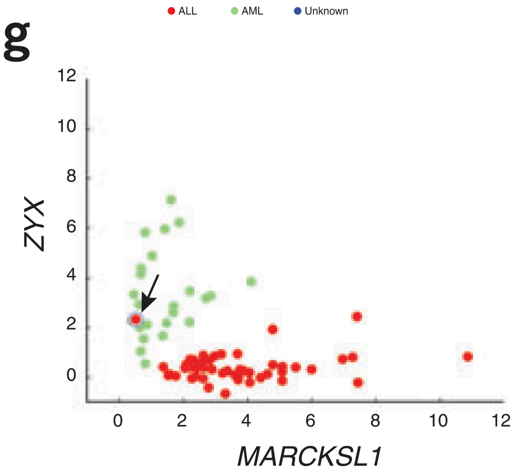
- So far, we have assumed that our data set is **linearly separable**.
- **Not** all the data sets are linearly separable and this will affect us in **two ways**.



Source: [1] Figure 1g

Linearly separable?

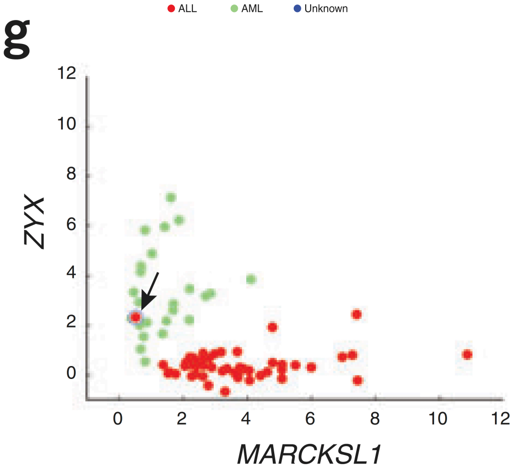
- So far, we have assumed that our data set is **linearly separable**.
- **Not** all the data sets are linearly separable and this will affect us in **two ways**.
- First, we might want to be able to allow for a **small number of classification errors**.



Source: [1] Figure 1g

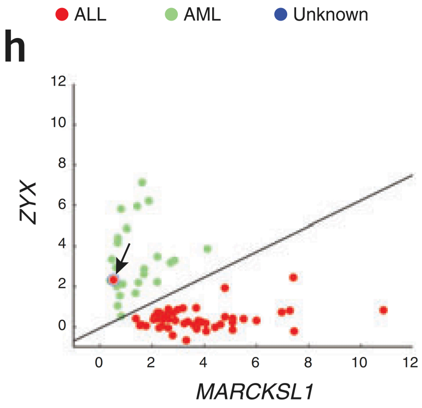
Linearly separable?

- So far, we have assumed that our data set is **linearly separable**.
- **Not** all the data sets are linearly separable and this will affect us in **two ways**.
- First, we might want to be able to allow for a **small number of classification errors**.
  - These might actually represent **errors in our training set**.



Source: [1] Figure 1g

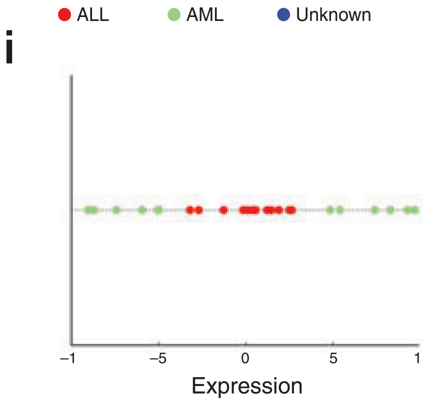
Soft margin



Source: [1] Figure 1h

- A user defined parameter, the **soft margin** (C), controls how many misclassification errors are allowed.

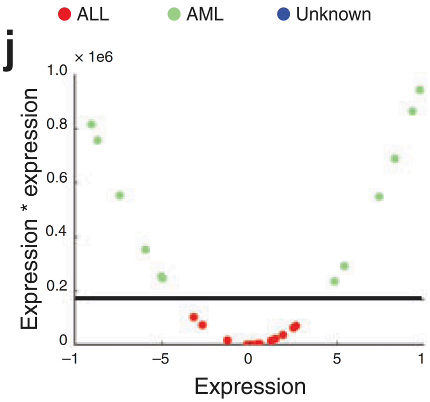
Linearly separable? (Take 2) — one feature



Source: [1] Figure 1i

- **No** single point can separate the two classes!

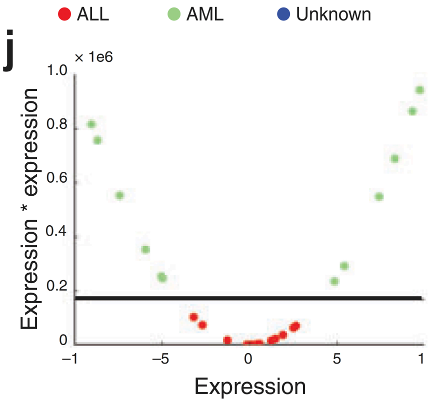
Kernel function



Source: [1] Figure 1j

- Adding a **new dimension** to our data.

Kernel function



Source: [1] Figure 1j

- Adding a **new dimension** to our data.
- Here, simply taking the square values of our feature.

Kernel function

- Informally, a **kernel function** can be seen as a mechanism to map data to a **higher dimensionality space**.

Kernel function

- Informally, a **kernel function** can be seen as a mechanism to map data to a **higher dimensionality space**.
- With some luck, intuition, intelligence or experience, one is able to find a **kernel function** such that the data is **linearly separable** in the resulting space.

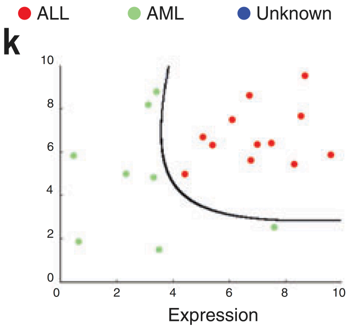
Kernel function

- Informally, a **kernel function** can be seen as a mechanism to map data to a **higher dimensionality space**.
- With some luck, intuition, intelligence or experience, one is able to find a **kernel function** such that the data is **linearly separable** in the resulting space.
- "It is possible to prove that, **for any given data set** with consistent labels (where consistent simply means that the data set does not contain two identical objects with opposite labels) **there exists a kernel function** that will **allow the data to be linearly separated**." [1]

Kernels

- Linear
- Polynomial
- Gaussian RBF (radial basis function)
- Sigmoid

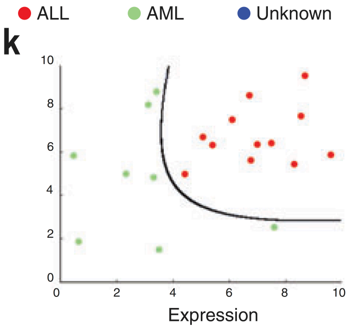
Kernel function



Source: [1] Figure 1k

- The result of projecting up the two-dimensional data into a four-dimensional space.

Kernel function



Source: [1] Figure 1k

- The result of projecting up the two-dimensional data into a four-dimensional space.
- Then projecting from the four-dimensional space down to a two-dimensional space (the curved line).

Errors/soft margin/kernel function

- "It is possible to prove that, **for any given data set** with consistent labels (where consistent simply means that the data set does not contain two identical objects with opposite labels) **there exists a kernel function** that will **allow the data to be linearly separated**." [1]

Errors/soft margin/kernel function

- “It is possible to prove that, **for any given data set** with consistent labels (where consistent simply means that the data set does not contain two identical objects with opposite labels) **there exists a kernel function** that will **allow the data to be linearly separated**.” [1]
- The above suggest that **we do not actually need the concept of soft margin**!

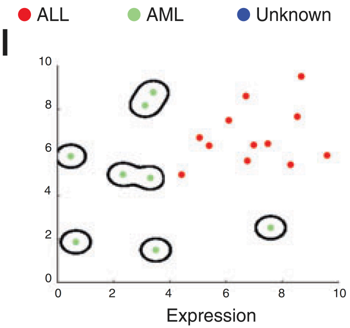
Errors/soft margin/kernel function

- "It is possible to prove that, **for any given data set** with consistent labels (where consistent simply means that the data set does not contain two identical objects with opposite labels) **there exists a kernel function** that will **allow the data to be linearly separated**." [1]
- The above suggest that **we do not actually need the concept of soft margin**!
- Simply project the data into higher a higher dimensionality space where it will be separable!

Errors/soft margin/kernel function

- “It is possible to prove that, **for any given data set** with consistent labels (where consistent simply means that the data set does not contain two identical objects with opposite labels) **there exists a kernel function** that will **allow the data to be linearly separated**.” [1]
- The above suggest that **we do not actually need the concept of soft margin**!
- Simply project the data into higher a higher dimensionality space where it will be separable!
- **Sadly**, it is not that simple!

Overfitting



Source: [1] Figure 1l

- The mapping to **higher-dimensional space** can create a **complex decision boundary** and **overfitting**.

Summary

- Just like **logistic regression**, **support vector machines** are learning the parameters of a **linear decision boundary** separating the data in **two classes**.

Summary

- Just like **logistic regression**, **support vector machines** are learning the parameters of a **linear decision boundary** separating the data in **two classes**.
- However, **SVM** algorithms also rely on the concept of **maximum margin hyperplane** as a mechanism to lower generalization errors.

Summary

- Just like **logistic regression**, **support vector machines** are learning the parameters of a **linear decision boundary** separating the data in **two classes**.
- However, **SVM** algorithms also rely on the concept of **maximum margin hyperplane** as a mechanism to lower generalization errors.
- In order to handle a **small number of classification errors**, the algorithms introduce the concept of **soft margin**.

Summary

- Just like **logistic regression**, **support vector machines** are learning the parameters of a **linear decision boundary** separating the data in **two classes**.
- However, **SVM** algorithms also rely on the concept of **maximum margin hyperplane** as a mechanism to lower generalization errors.
- In order to handle a **small number of classification errors**, the algorithms introduce the concept of **soft margin**.
- Finally, because the data is **not always linearly separable**, these algorithms project the data to **higher dimensions** using a **kernel function**.

Multiclass SVM

- **One** of the simplest ways to transform a **binary classification** problem into **multiclass** is to solve K **binary classification** problems in the following way:

Multiclass SVM

- **One** of the simplest ways to transform a **binary classification** problem into **multiclass** is to solve K **binary classification** problems in the following way:
  - Build a classifier to distinguish between class **1** against the rest.

Multiclass SVM

- **One** of the simplest ways to transform a **binary classification** problem into **multiclass** is to solve K **binary classification** problems in the following way:
  - Build a classifier to distinguish between class **1** against the rest.
  - Build a classifier to distinguish between class **2** against the rest.

Multiclass SVM

- **One** of the simplest ways to transform a **binary classification** problem into **multiclass** is to solve K **binary classification** problems in the following way:
  - Build a classifier to distinguish between class **1** against the rest.
  - Build a classifier to distinguish between class **2** against the rest.
  - ...

Multiclass SVM

- **One** of the simplest ways to transform a **binary classification** problem into **multiclass** is to solve K **binary classification** problems in the following way:
  - Build a classifier to distinguish between class **1** against the rest.
  - Build a classifier to distinguish between class **2** against the rest.
  - ...
  - Build a classifier to distinguish between Class **K** against the best.

Multiclass SVM

- **One** of the simplest ways to transform a **binary classification** problem into **multiclass** is to solve K **binary classification** problems in the following way:
  - Build a classifier to distinguish between class **1** against the rest.
  - Build a classifier to distinguish between class **2** against the rest.
  - ...
  - Build a classifier to distinguish between Class **K** against the best.
- In other words, examples of **class** k (**label** $=$ **1**) are the positive examples and all the other examples are the negative examples (**label** $=$ **-1**), $\forall k \in 1 \dots K$.

Implementation

Worked example on kaggle

- https://www.kaggle.com/crawford/gene-expression
- https://www.kaggle.com/varimp/gene-expression-classification

sklearn.svm.SVC

scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

```
svm_param_grid = {'C': [0.1, 1, 10, 100],
                  'kernel': ['linear', 'rbf', 'poly']}

svm_grid = GridSearchCV(SVC(), svm_param_grid, cv=3)

svm_grid.fit(X_train, y_train)

print("Best Parameters:\n", svm_grid.best_params_)

best_svc = svm_grid.best_estimator_

svm_pred = best_svc.predict(X_test)
```

Source: `https://www.kaggle.com/varimp/gene-expression-classification`

Complexity

- Grows **quadratically** with the **number of examples**.
- **Linear time** approximations exist.
- Handles **millions of examples**.

Prologue

Kernel methods

- As we will see in the next lecture, **kernels** can be defined for inputs that are not vectors.

Summary

- Just like **logistic regression**, **support vector machines** are learning the parameters of a linear decision boundary separating the data in two classes.

Summary

- Just like **logistic regression**, **support vector machines** are learning the parameters of a linear decision boundary separating the data in two classes.
- However, **SVM** algorithms also rely on the concept of **maximum margin hyperplane** as a mechanism to lower generalization errors.

Summary

- Just like **logistic regression**, **support vector machines** are learning the parameters of a linear decision boundary separating the data in two classes.
- However, **SVM** algorithms also rely on the concept of **maximum margin hyperplane** as a mechanism to lower generalization errors.
- In order to handle a small number of classification errors, the algorithms introduce the concept of **soft margin**.

Summary

- Just like **logistic regression**, **support vector machines** are learning the parameters of a linear decision boundary separating the data in two classes.
- However, **SVM** algorithms also rely on the concept of **maximum margin hyperplane** as a mechanism to lower generalization errors.
- In order to handle a small number of classification errors, the algorithms introduce the concept of **soft margin**.
- Finally, because the data is **not always linearly separable**, these algorithms project the data to **higher dimensions** using a **kernel function**.

Summary

- Just like **logistic regression**, **support vector machines** are learning the parameters of a linear decision boundary separating the data in two classes.
- However, **SVM** algorithms also rely on the concept of **maximum margin hyperplane** as a mechanism to lower generalization errors.
- In order to handle a small number of classification errors, the algorithms introduce the concept of **soft margin**.
- Finally, because the data is **not always linearly separable**, these algorithms project the data to **higher dimensions** using a **kernel function**.
- **Support vector machines** have a **strong foundation**.

Summary

- Just like **logistic regression**, **support vector machines** are learning the parameters of a linear decision boundary separating the data in two classes.
- However, **SVM** algorithms also rely on the concept of **maximum margin hyperplane** as a mechanism to lower generalization errors.
- In order to handle a small number of classification errors, the algorithms introduce the concept of **soft margin**.
- Finally, because the data is **not always linearly separable**, these algorithms project the data to **higher dimensions** using a **kernel function**.
- **Support vector machines** have a **strong foundation**.
- **Empirical results** show that their performance is excellent.

Next module

- **Kernel Methods**

References

William S Noble.
What is a support vector machine?
Nat Biotechnol, 24(12):1565–7, Dec 2006.

T R Golub, D K Slonim, P Tamayo, C Huard, M Gaasenbeek, J P Mesirov, H Coller, M L Loh, J R Downing, M A Caligiuri, C D Bloomfield, and E S Lander.
Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.
Science, 286(5439):531–7, Oct 1999.

Michael Molla, Michael Waddell, David Page, and Jude W. Shavlik.
Using machine learning to design and interpret gene-expression microarrays.
AI Magazine, 25(1):23–44, 2004.

Aurélien Géron.
Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow.
O'Reilly Media, 2nd edition, 2019.

Marcel **Turcotte**
Marcel.Turcotte@uOttawa.ca

School of Electrical Engineering and **Computer Science** (EECS)
University of Ottawa