DS-GA 3001.007
Introduction to Machine Learning

Lecture 12

Features and Labels – Working with Kernels

Determining features and labels to extend linear regression and linear classification

DS-GA 3001.007
Introduction to Machine Learning

Lecture 12

Features and Labels – Working with Kernels

Center for Data Science
NYU

Determining features and labels to extend linear regression and linear classification

DS-GA 3001.007
Introduction to Machine Learning

Relationships between features

Lecture 12

Features and Labels – Working with Kernels

Announcements

- Homework
  - Homework 5 due **Tuesday November 26** at 11:59pm
- Project
  - Milestone due **Thursday November 28** at 11:59pm
  - Background and Plans
- Labs
  - Submit on Jupyter Hub under Assignments tab

Refer to weekly agenda for more information

Announcements

- Homework
  - Homework 5 due **Tuesday November 26** at 11:59pm
- Project
  - Milestone due **Thursday November 28** at 11:59pm
  - Background and Plans
- Labs
  - Submit on Jupyter Hub under Assignments tab

Due on Mondays

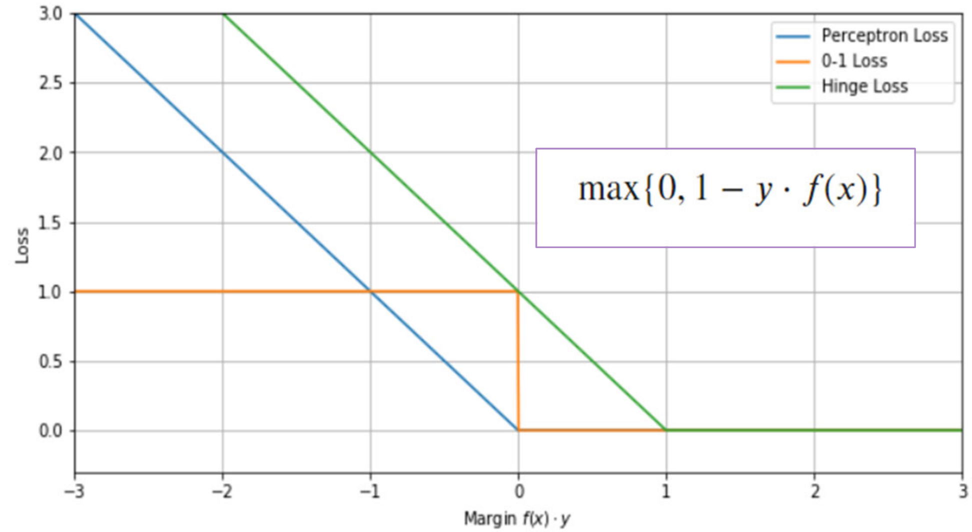
Review

- Support Vector Machine
  - Linear Classifier with hypothesis space

$$\mathcal{F} = \{f(x) = w^T x + b \mid w \in \mathbf{R}^d, b \in \mathbf{R}\}$$

  - Hinge Loss not Perceptron Loss to capture confidence of classification with margin
  - Use l_2 regularization term

$$\frac{1}{2}||w||^2 + \frac{c}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i\left[w^T x_i + b\right]\right)$$



Review

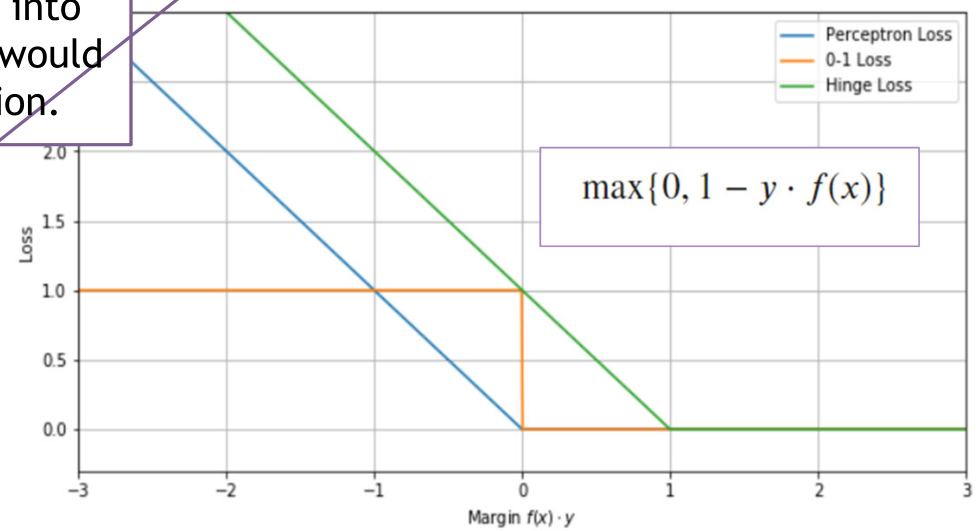
Could incorporate offset into weights. However, offset would shrink from regularization.

- Support Vector Machine
  - Linear Classifier with hypothesis space

$$\mathcal{F} = \{f(x) = w^T x + b \mid w \in \mathbf{R}^d, b \in \mathbf{R}\}$$

  - Hinge Loss not Perceptron Loss to capture confidence of classification with margin
  - Use l_2 regularization term

$$\frac{1}{2}||w||^2 + \frac{c}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i\left[w^T x_i + b\right]\right)$$



$$\max\{0, 1 - y \cdot f(x)\}$$

To avoid regularizing the offset, choose the corresponding feature to be a large number instead of 1

Review

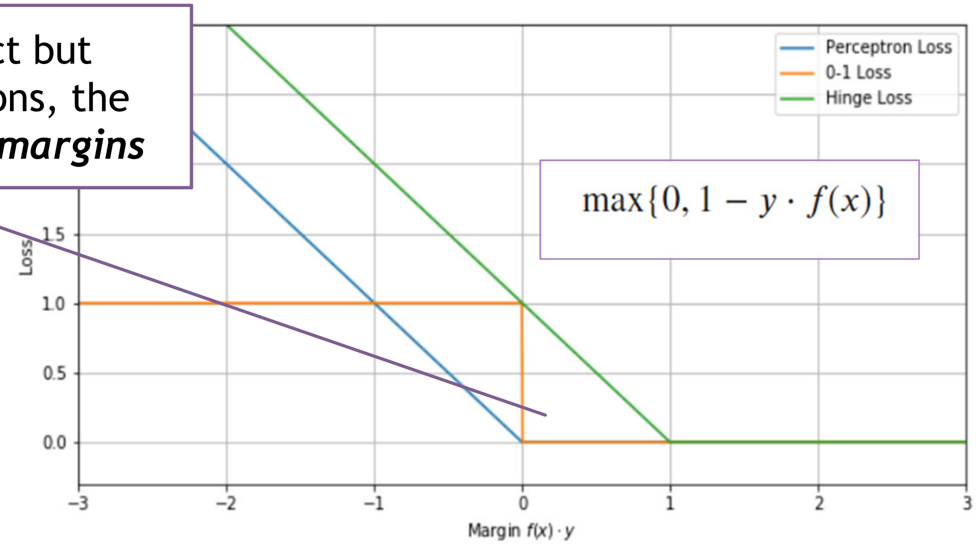
Since penalizes correct but unconfident classifications, the model will give us ***large margins***

- Support Vector Machine
  - Linear Classifier with hypothesis space

$$\mathcal{F} = \{f(x) = w^T x + b \mid w \in \mathbf{R}^d, b \in \mathbf{R}\}$$

  - Hinge Loss not Perceptron Loss to capture confidence of classification with margin
  - Use l_2 regularization term

$$\frac{1}{2}||w||^2 + \frac{c}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i\left[w^T x_i + b\right]\right)$$



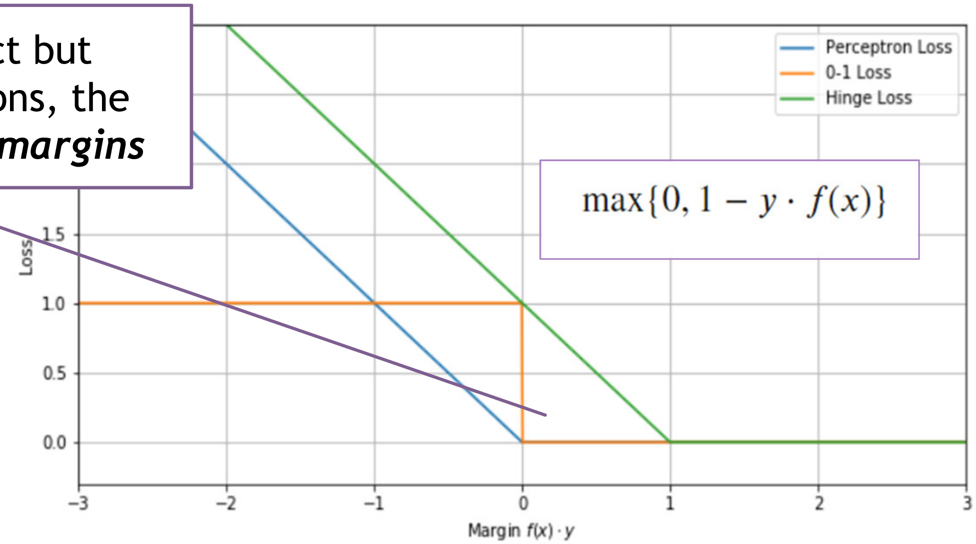
$$\max\{0, 1 - y \cdot f(x)\}$$

Review

Since penalizes correct but unconfident classifications, the model will give us ***large margins***

- Support Vector Machine
  - Linear Classifier with hypothesis space

$$\mathcal{F} = \{ f(x) = w^T x + b \mid w \in \mathbf{R}^d, b \in \mathbf{R} \}$$



$$\max\{0, 1 - y \cdot f(x)\}$$

  - Hinge Loss not Perceptron Loss to capture confidence of classification with margin
  - Use l_2 regularization term

$$\frac{1}{2}||w||^2 + \frac{c}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i \left[w^T x_i + b\right]\right)$$

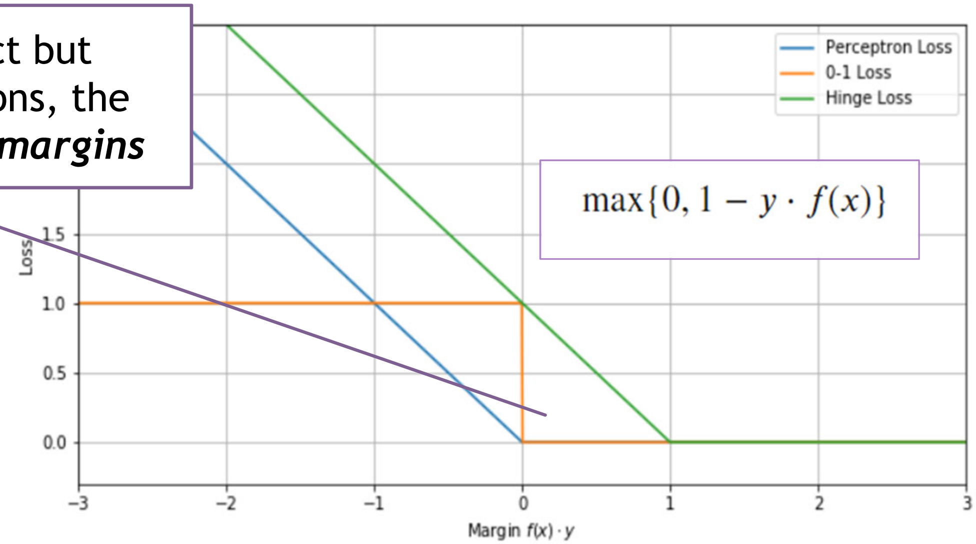
Measures accuracy of classification

Review

Since penalizes correct but unconfident classifications, the model will give us ***large margins***

- Support Vector Machine
  - Linear Classifier with hypothesis space

$$\mathcal{F} = \{f(x) = w^T x + b \mid w \in \mathbf{R}^d, b \in \mathbf{R}\}$$



$$\max\{0, 1 - y \cdot f(x)\}$$

  - Hinge Loss not Perceptron Loss to capture confidence of classification with margin
  - Use l_2 regularization term

$$\frac{1}{2}||w||^2 + \frac{c}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i \left[w^T x_i + b\right]\right)$$

Measures confidence of classification

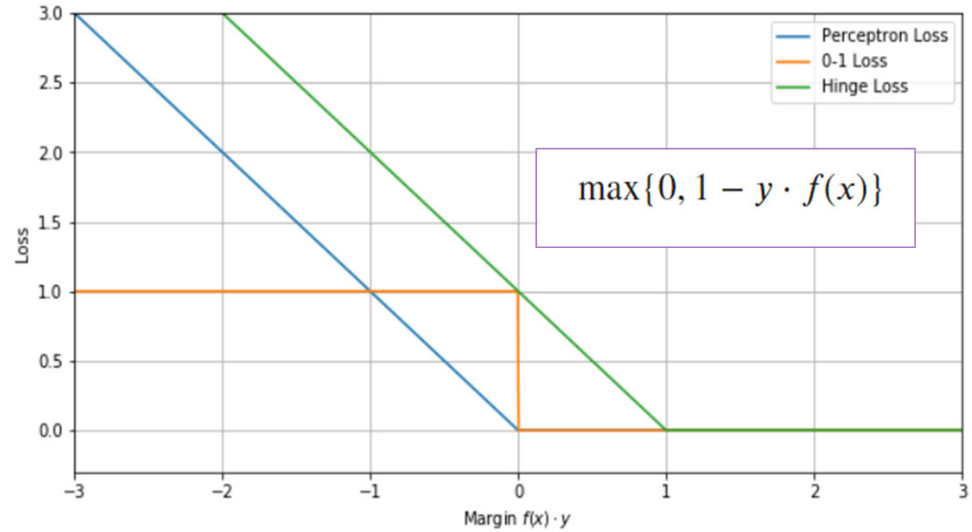
Measures accuracy of classification

Review

- Support Vector Machine
  - Linear Classifier with hypothesis space

$$\mathcal{F} = \{f(x) = w^T x + b \mid w \in \mathbf{R}^d, b \in \mathbf{R}\}$$

  - Hinge Loss not Perceptron Loss to capture confidence of classification with margin
  - Use l_2 regularization term



$$\max\{0, 1 - y \cdot f(x)\}$$

$$\frac{1}{2}||w||^2 + \frac{c}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i\left[w^T x_i + b\right]\right)$$

Measures confidence of classification

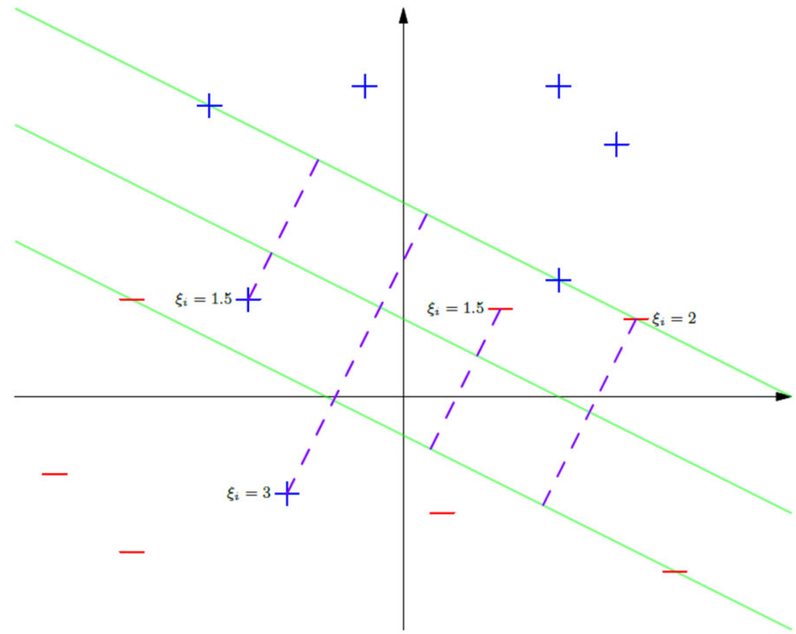
Measures accuracy of classification

Review

sup means max

- Primal Problem and Dual Problem
  - Formulated constrained minimization problem
  - We combined objective and constraint to form Lagrangian
  - Switching order of minimum and maximum we obtained

$$
\begin{aligned}
&\sup_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_j^T x_i \\
&\text{s.t.} \sum_{i=1}^{n} \alpha_i y_i = 0 \\
&\qquad \alpha_i \in \left[0, \frac{c}{n}\right].
\end{aligned}
$$



Review

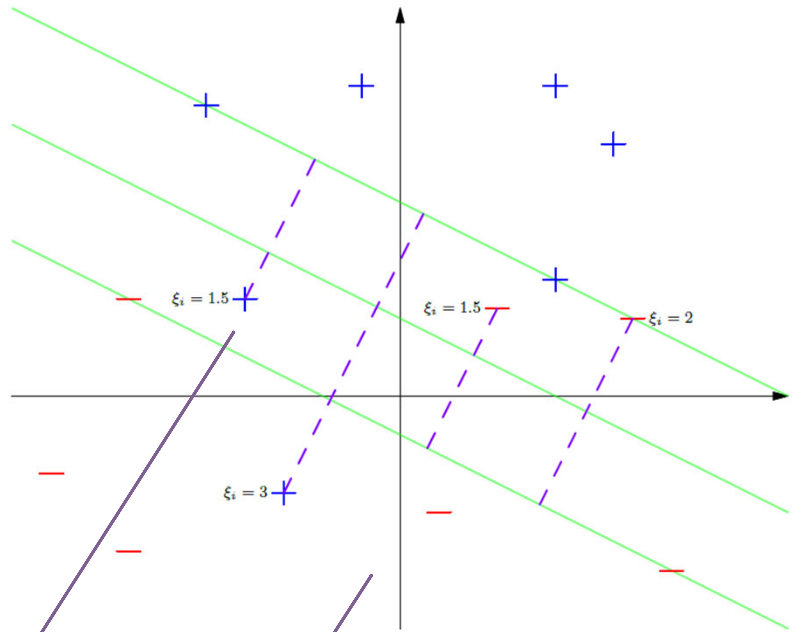
- Primal Problem and Dual Problem
  - Formulated constrained minimization problem
  - We combined objective and constraint to form Lagrangian
  - Switching order of minimum and maximum we obtained

sup means max

$$\sup_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_j^T x_i$$

$$\text{s.t.} \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\alpha_i \in \left[0, \frac{c}{n}\right].$$

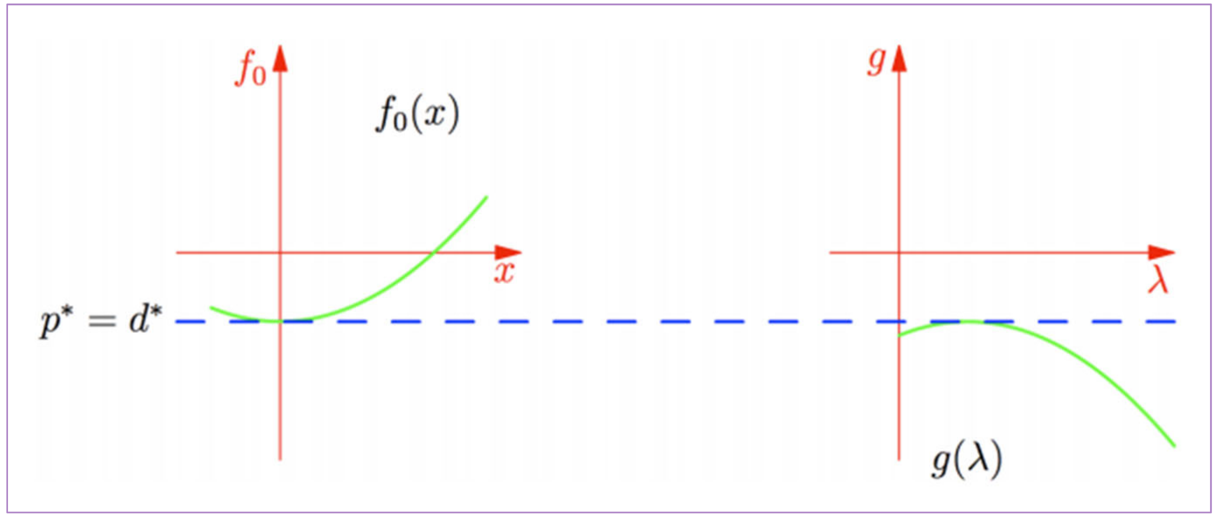


Soft Margin and Hard Margin Formulations of SVM

Support vectors are those vectors that impact the separating plane

Review

- Primal Problem and Dual Problem
  - Equality between the primal problem and dual problem
  - First Order Conditions and Complementary Slackness showed relationships between dual variables and constraints



$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

Review

- Primal Problem and Dual Problem
  - Equality between the primal problem and dual problem
  - First Order Conditions and Complementary Slackness showed relationships between dual variables and constraints

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

$$\begin{aligned}
\alpha_i^* = 0 \quad &\implies \quad y_i f^*(x_i) \geq 1 \\
\alpha_i^* \in \left(0, \frac{c}{n}\right) \quad &\implies \quad y_i f^*(x_i) = 1 \\
\alpha_i^* = \frac{c}{n} \quad &\implies \quad y_i f^*(x_i) \leq 1
\end{aligned}$$

$$\begin{aligned}
y_i f^*(x_i) < 1 \quad &\implies \quad \alpha_i^* = \frac{c}{n} \\
y_i f^*(x_i) = 1 \quad &\implies \quad \alpha_i^* \in \left[0, \frac{c}{n}\right] \\
y_i f^*(x_i) > 1 \quad &\implies \quad \alpha_i^* = 0
\end{aligned}$$

Review

Only positive values are support vectors

- Primal Problem and Dual Problem
  - Equality between the primal problem and dual problem
  - First Order Conditions and Complementary Slackness showed relationships between dual variables and constraints

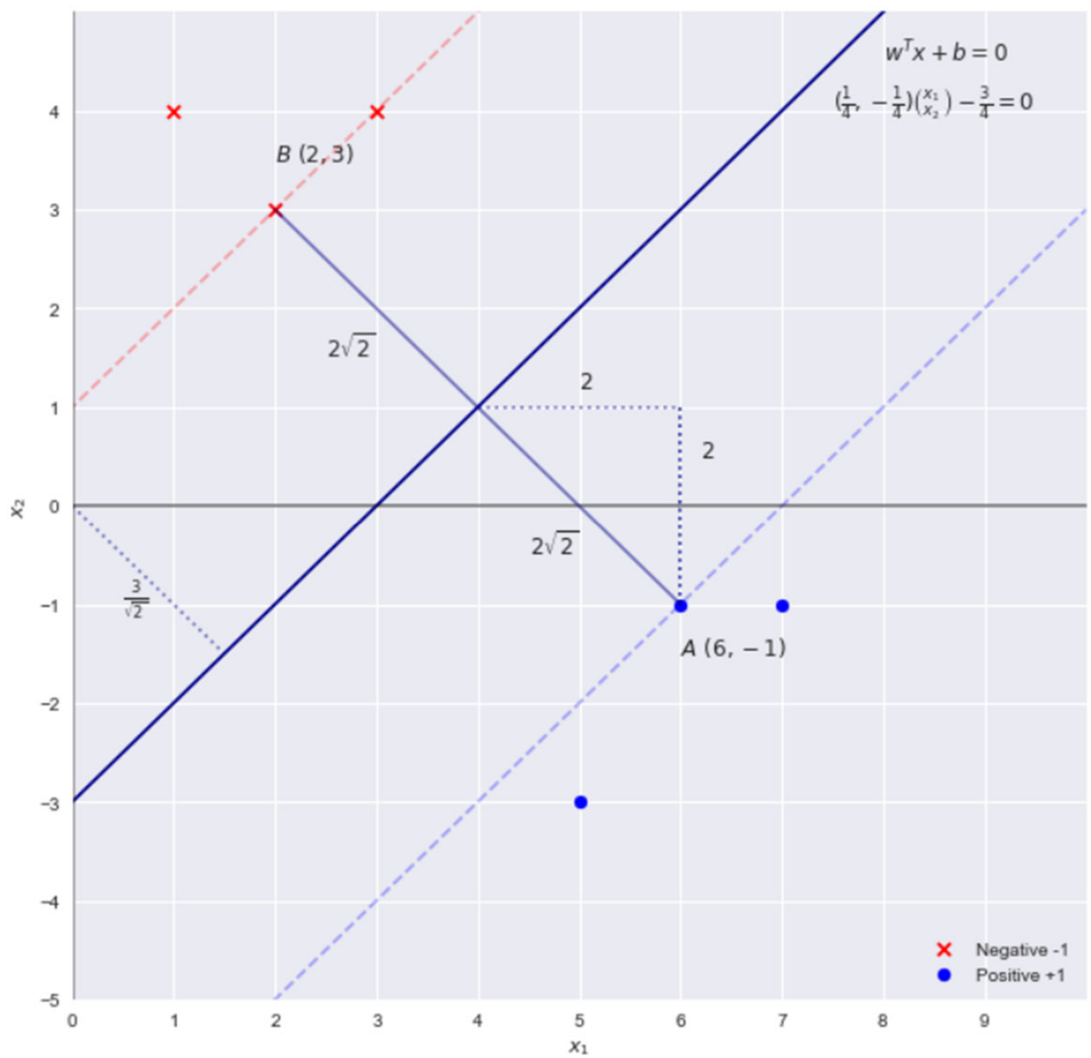
$$
\begin{aligned}
\alpha_i^* = 0 \quad &\Longrightarrow \quad y_i f^*(x_i) \geq 1 \\
\alpha_i^* \in \left(0, \frac{c}{n}\right) \quad &\Longrightarrow \quad y_i f^*(x_i) = 1 \\
\alpha_i^* = \frac{c}{n} \quad &\Longrightarrow \quad y_i f^*(x_i) \leq 1
\end{aligned}
$$

$$
\begin{aligned}
y_i f^*(x_i) < 1 \quad &\Longrightarrow \quad \alpha_i^* = \frac{c}{n} \\
y_i f^*(x_i) = 1 \quad &\Longrightarrow \quad \alpha_i^* \in \left[0, \frac{c}{n}\right] \\
y_i f^*(x_i) > 1 \quad &\Longrightarrow \quad \alpha_i^* = 0
\end{aligned}
$$

$$
w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i
$$

Some points in training set at minimal distance to separating plane are not support vectors

Review



$w^T x + b = 0$
$(\frac{1}{4}, -\frac{1}{4})\binom{x_1}{x_2} - \frac{3}{4} = 0$
$B\ (2, 3)$
$2\sqrt{2}$
2
2
$2\sqrt{2}$
$\frac{3}{\sqrt{2}}$
$A\ (6, -1)$
Negative -1
Positive +1
x_1
x_2

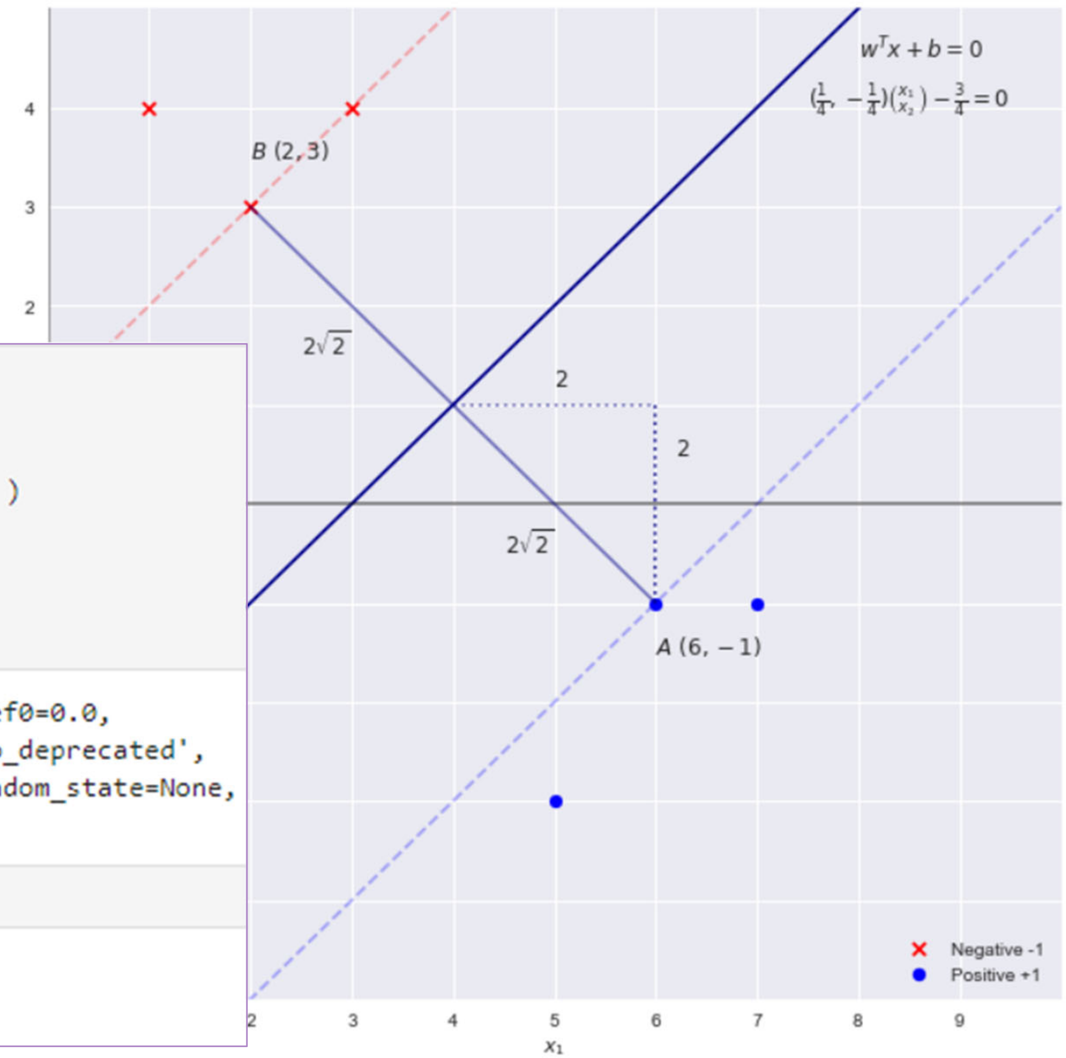
Review

```python
import numpy as np
from sklearn.svm import SVC

X = np.array([[3,4],[1,4],[2,3],[6,-1],[7,-1],[5,-3]] )
y = np.array([-1,-1, -1, 1, 1 , 1 ])

clf = SVC(C = 1e5, kernel = 'linear')
clf.fit(X, y)
```

```
SVC(C=100000.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='linear', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
```

```python
clf.support_vectors_
```

```
array([[ 2.,  3.],
       [ 6., -1.]])
```



Review

$$w = [1, -1] \;\; b = -3$$

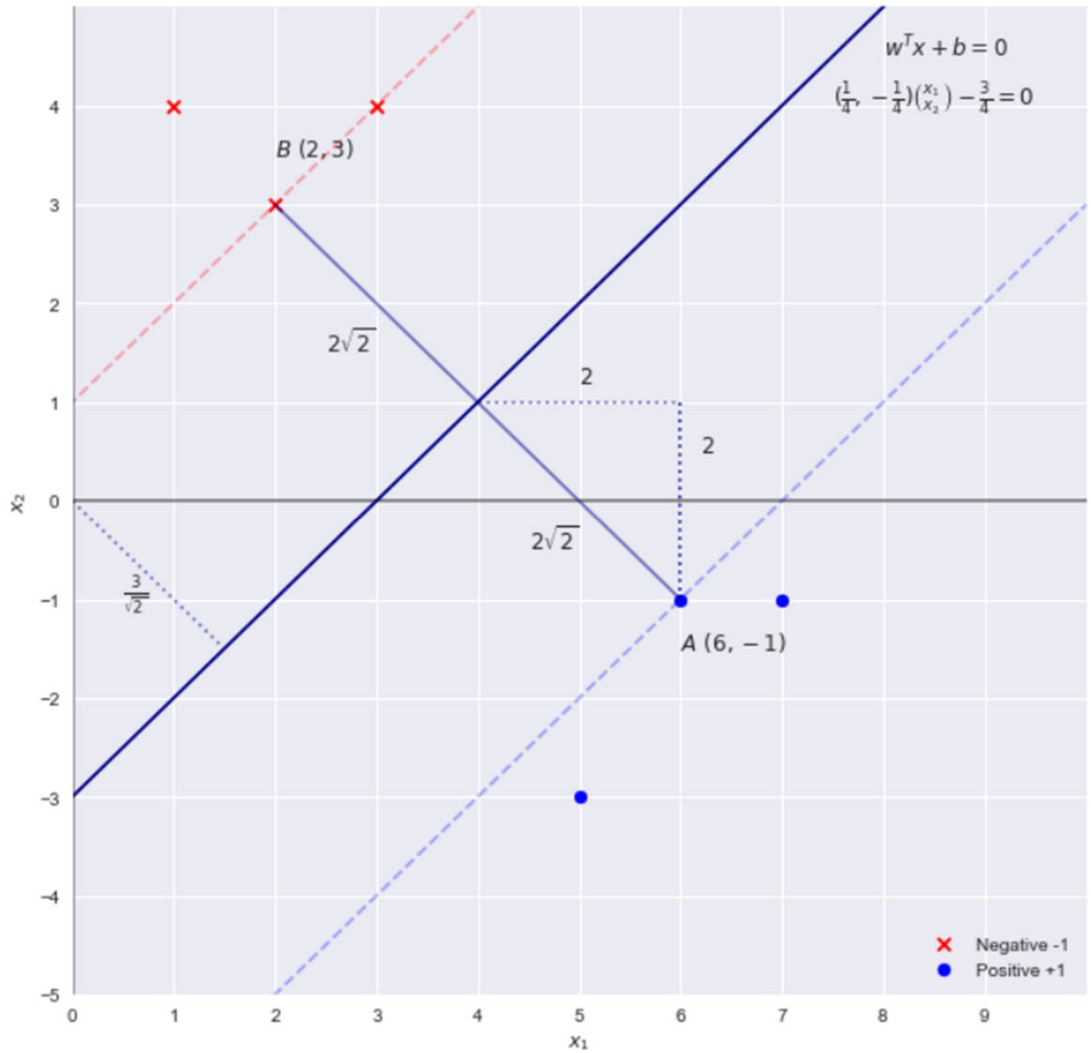
$$cx_1 - xc_2 - 3c = 0$$

$$w = [c, -c] \;\; b = -3c$$

$$\frac{2}{||w||} = 4\sqrt{2}$$

$$\frac{2}{\sqrt{2}c} = 4\sqrt{2}$$

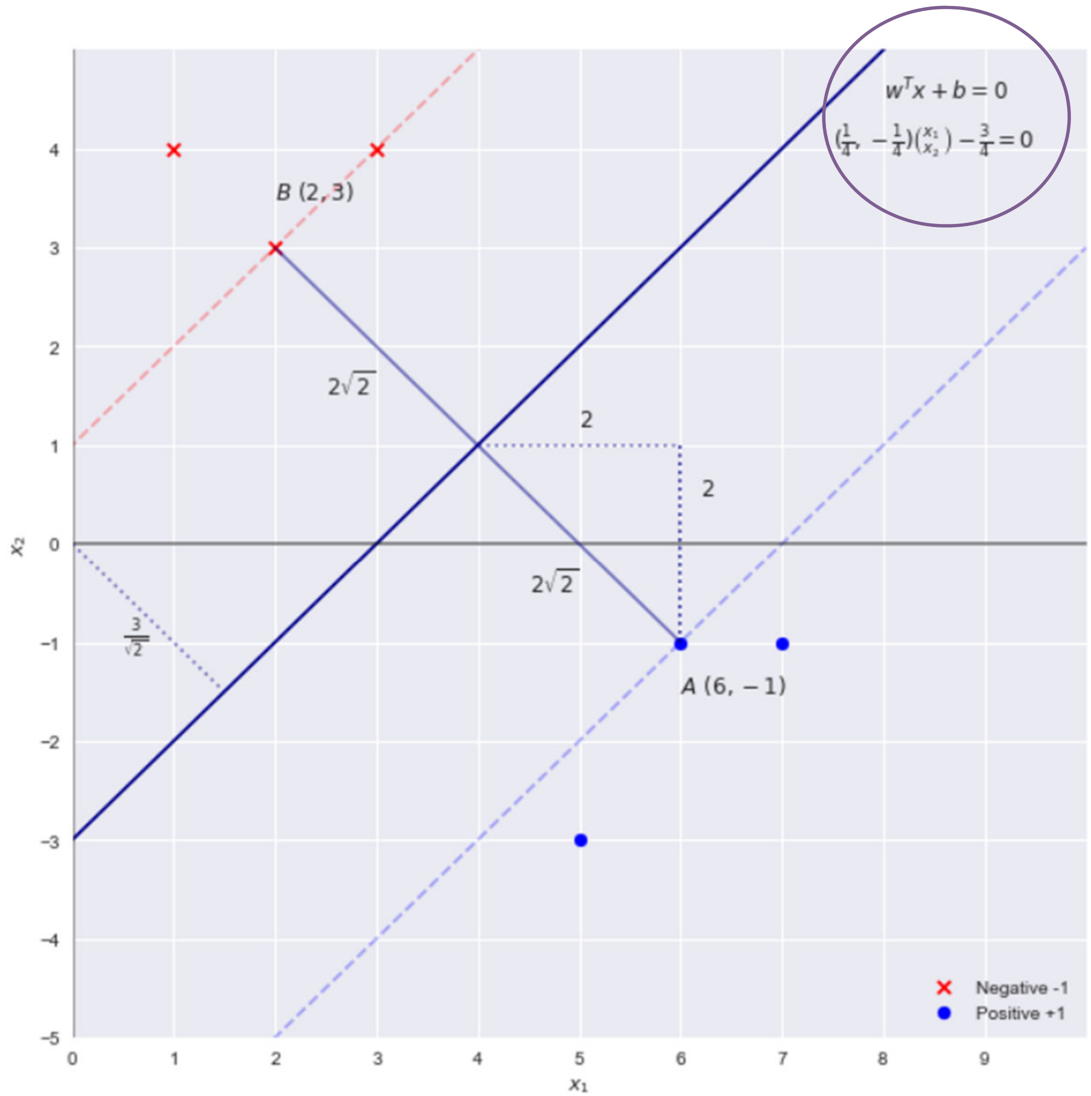
$$c = \frac{1}{4}$$



Review

$$w = [1, -1] \;\; b = -3$$

$$w = [\frac{1}{4}, -\frac{1}{4}] \;\; b = -\frac{3}{4}$$



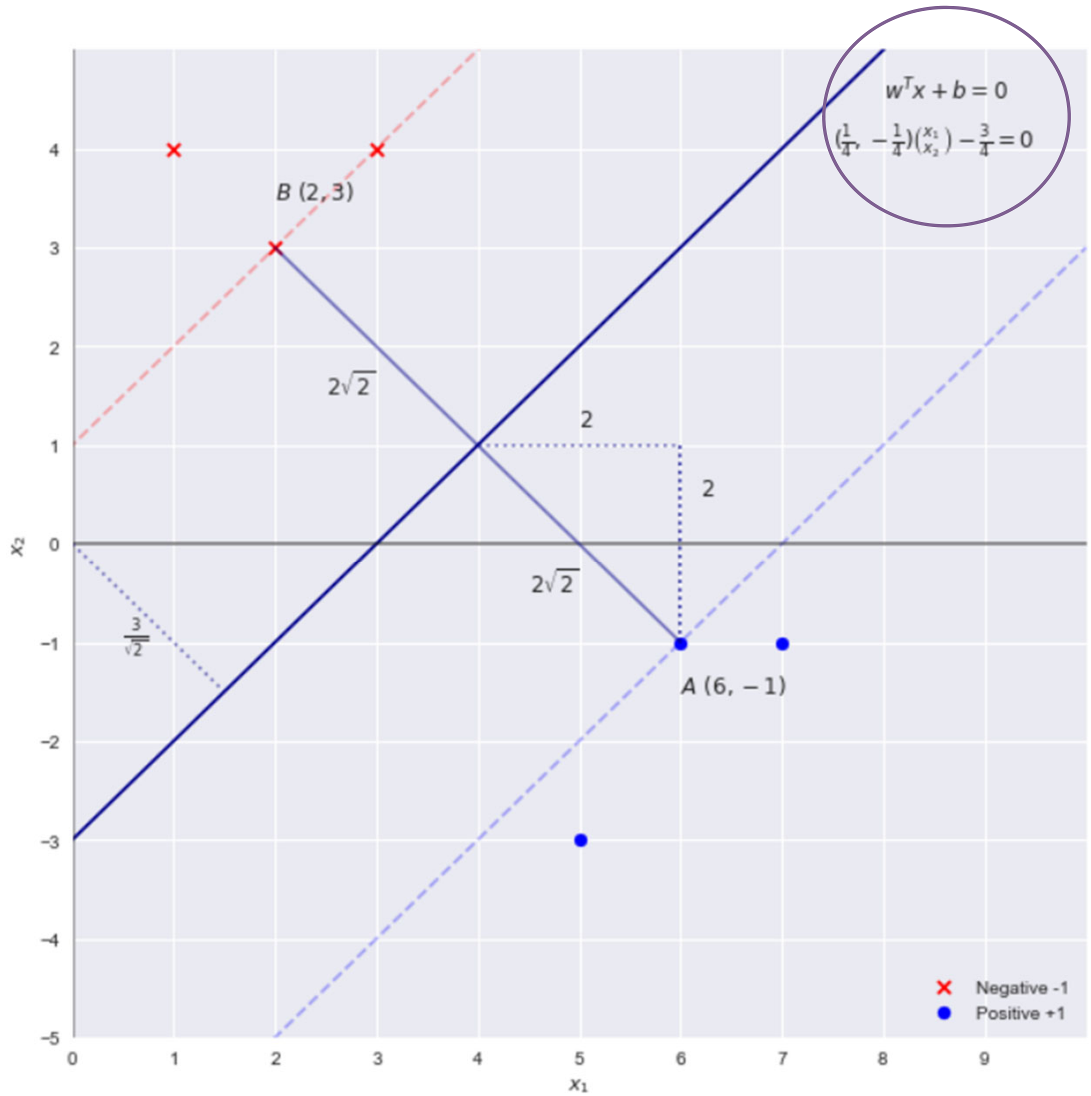
Review

$$w = \sum_{i}^{m} \alpha_i y^{(i)} x^{(i)}$$

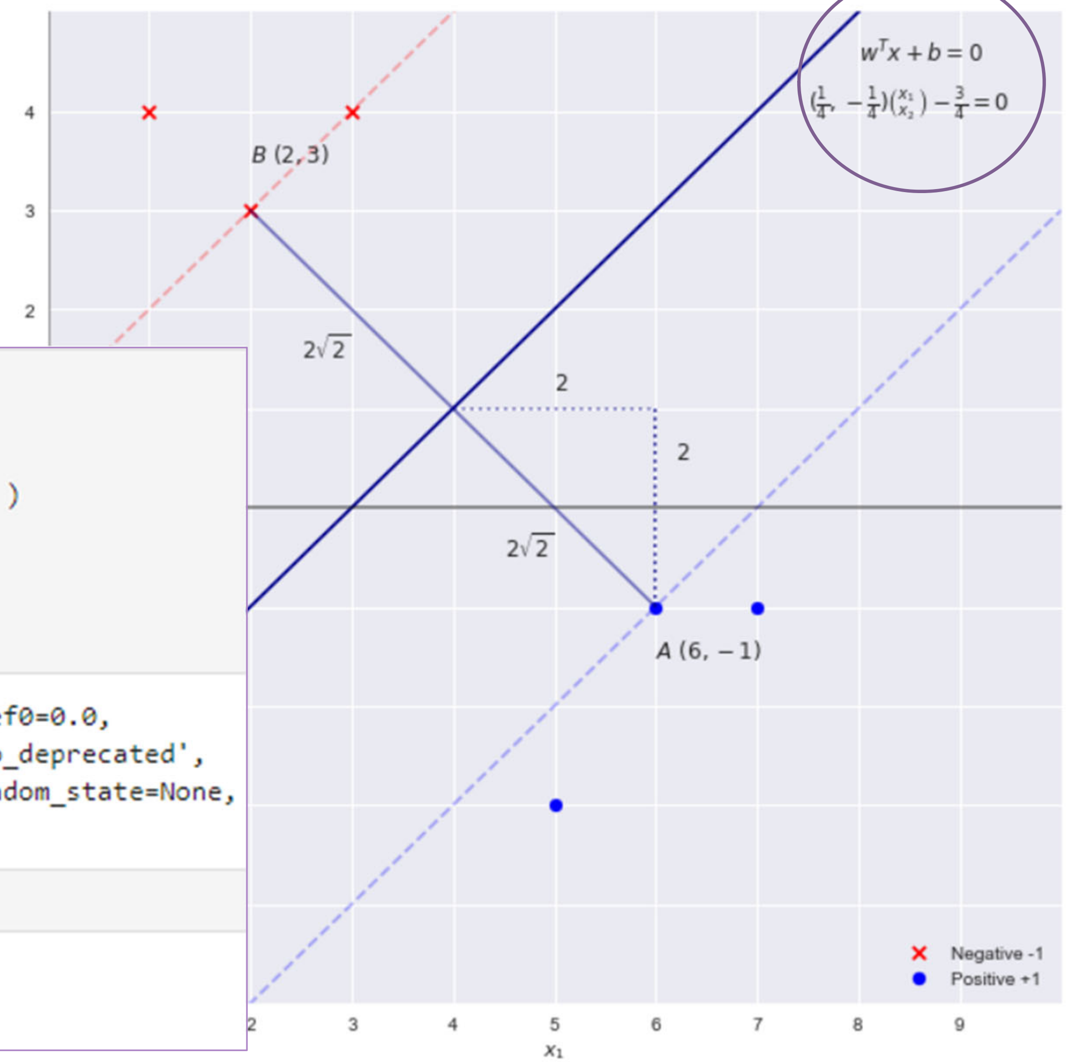
$$\sum_{i}^{m} \alpha_i y^{(i)} = 0$$

$$\begin{bmatrix} 6\alpha_1 - 2\alpha_2 - 3\alpha_3 \\ -1\alpha_1 - 3\alpha_2 - 4\alpha_3 \\ 1\alpha_1 - 1\alpha_2 - 1\alpha_3 \end{bmatrix} = \begin{bmatrix} 1/4 \\ -1/4 \\ 0 \end{bmatrix}$$

$$\alpha = \begin{bmatrix} 1/16 \\ 1/16 \\ 0 \end{bmatrix}$$



Review




```
import numpy as np
from sklearn.svm import SVC

X = np.array([[3,4],[1,4],[2,3],[6,-1],[7,-1],[5,-3]] )
y = np.array([-1,-1, -1, 1, 1 , 1 ])

clf = SVC(C = 1e5, kernel = 'linear')
clf.fit(X, y)
```

```
SVC(C=100000.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='linear', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
```

```
clf.support_vectors_
```

```
array([[ 2.,  3.],
       [ 6., -1.]])
```

Review

Application of SVM

- Transfer Learning
  - Suppose we want to build spam classifier like HW1 for Alice.
  - Alice has few labelled emails. Bob has many labelled emails.
  - **Question**: Assuming that Alice and Bob have related notion of spam, how can we use Bob's data / classifier?

Review

Application of SVM

- **Question:**
  - Suppose we want to build spam classifier like HW1 for Alice.
  - Alice has few labelled emails. Bob has many labelled emails.
  - Assuming that Alice and Bob have related notion of spam, how can we use Bob's data / classifier?
- **Ideas:**
  - Average weights between Alice and Bob?
  - Combine emails. Duplicating emails of Alice
  - **Transfer Learning:** Modify the SVM objective to use Bob's weights

Review

$$\min_{\mathbf{w}_d, b_d} \frac{C}{|D_d|} \sum_{\mathbf{x}, y \in D_d} \max(0, 1 - y(\mathbf{w}_d^T \mathbf{x} + b_d)) + \frac{1}{2} \|\mathbf{w}_d - \mathbf{w}_r\|^2$$

Application of SVM

- **Question:**
  - Suppose we want to build spam classifier like HW1 for Daniel.
  - Daniel has few labelled emails. Richard has many labelled emails.
  - Assuming that Daniel and Richard have related notion of spam, how can we use Richard's data / classifier?
- **Ideas:**
  - Average weights between Daniel and Richard?
  - Combine emails. Duplicating emails of Daniel
  - **Transfer Learning:** Modify the SVM objective to use Richard's weights

Review

- The weights for solution are linear combination of points in training set

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

- Many of the dual variables are 0. So weight is sparse in the data.

Key Points of SVM

Relationship between weights and data is common. For example, Perceptron with initial weights 0

Key Points of SVM

Review

- The weights for solution are linear combination of points in training set

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

Relationship between weights and data is common. For example, Perceptron with initial weights 0

- Many of the dual variables are 0. So weight is sparse in the data.

- Since the dual variables are bounded between 0 and c/n, the hyperparameter controls the size.
- Support vectors impact separating plane. Dropping any will change the plane

Review

- Support Vector Machines have three components
  - Sparsity
  - Large Margin
  - Kernels
- Sparsity and Large Margin…
  - Stem from Hinge Loss instead of Perceptron loss
  - Prevent against Overfitting

Moreover sparsity allows us to interpret the weights in terms of the data.

Key Points of SVM

Review

- Support Vector Machines have three components
  - Sparsity
  - Large Margin
  - Kernels
- Sparsity and Large Margin…
  - Stem from Hinge Loss instead of Perceptron loss
  - Prevent against Overfitting

Kernels implicitly use a high dimensional space of features. However, only the relationship between features needed for calculation.

Key Points of SVM

- Kernels
  - Prevent against underfitting
  - Allow for use of many features without optimization error
  - We have been using the ***linear kernel***. We can reuse the same code for other kernels

Agenda

- Lesson
  - Features
    - Encoding text, images, recordings into numbers
  - Kernels
    - Relationships between features
  - Labels
- Demo
  - Features for Regression

Objectives

- Can we extend linear regression / classification with different features?
- Could we replace features with relationships between features in some algorithms?
- How can we study more than two categories for classification?
- **Readings**:
  - Shalev-Schwarz Chapter 16 (see 16.3 for modified SGD)
  - Murphy Chapter 14.5 (see 14.5.2.4 for multiclass classification)

Agenda

- Lesson
  - Features
    - Encoding text, images, recordings into numbers
  - Kernels
    - Relationships between features
  - Labels
- Demo
  - Features for Regression

Objectives

- **Can we extend linear regression / classification with different features?**
- Could we replace features with relationships between features in some algorithms?
- How can we study more than two categories for classification?
- **Readings**:
  - Shalev-Schwarz Chapter 16 (see 16.3 for modified SGD)
  - Murphy Chapter 14.5 (see 14.5.2.4 for multiclass classification)

Features

- Data Types
  - We work with
    - Text
    - Images
    - Recordings
  - These need to be translated into numbers for use in the model
- Fixed Size
  - Each feature has the same length. However, we should explore different sizes.
  - Should we have to determine the features from domain knowledge?

Features

- Data Types
  - We work with
    - Text
    - Images
    - Recordings
  - These need to be translated into numbers for use in the model
- Fixed Size
  - Each feature has the same length. However, we should explore different sizes.
  - Should we have to determine the features from domain knowledge?

Example

- Suppose we want to predict whether string is email.
  - Contains @ symbol - 0 or 1
  - Contains gmail, outlook, etc. - 0 or 1
  - Ends with .com or .edu - 0 or 1
  - …
- Rather than hard-code the ending, we can allow it to vary
  - aaa, aab, …, zzz
  - One-Hot Encode these features

Features

- How to avoid Approximation Error with Linear Models?
  - Linear regression and linear classification may not be able to fit the training set
  - Sometimes we have
    - Nonlinear trends in the data

Features

- How to avoid Approximation Error with Linear Models?
  - Linear regression and linear classification may not be able to fit the training set
  - Sometimes we have
    - Nonlinear trends in the data

Example

- Suppose we want to predict health from weight
  - Relationship between health and weight is not linear.
- Suppose the target weight for a certain person x is w.
  - Encode features as f(x). If f(x) = [1, weight(x)] then we cannot predict health with a linear model
  - Want $f(x) = [1, (weight(x) - w)^2]$
  - So we can use

$f(x) = [1, weight(x), weight(x)^2]$

Think action space is R for health score. Translate into label of healthy vs not healthy

Features

- How to avoid Approximation Error with Linear Models?
  - Linear regression and linear classification may not be able to fit the training set
  - Sometimes we have
    - Nonlinear trends in the data
    - Interaction between features

Features

- How to avoid Approximation Error with Linear Models?
  - Linear regression and linear classification may not be able to fit the training set
  - Sometimes we have
    - Nonlinear trends in the data
    - Interaction between features

Example

- Suppose we want to predict health from weight and height
  - Health determined by weight relative to height
- Suppose the health weight for height is
  - w = 52 + 1.9 (h - 60)
- Take feature encoding for person x to be

  $f(x) = (w(x) - (52 + 1.9 (h(x) - 60))^2$
- However, we could use

$f(x) = [1, h(x), w(x), h(x)^2, w(x)^2, h(x)w(x)]$

Features

- How to avoid Approximation Error with Linear Models?
  - Linear regression and linear classification may not be able to fit the training set
  - Sometimes we have
    - Nonlinear trends in the data
    - Interaction between features
    - Categorical to Numerical

Features

- How to avoid Approximation Error with Linear Models?
  - Linear regression and linear classification may not be able to fit the training set
  - Sometimes we have
    - Nonlinear trends in the data
    - Interaction between features
    - Categorical to Numerical

Example

- Suppose we want to predict health from weight and height for three different body types – A, B, C
- Suppose the healthy weight for height is
  - w = 52 + 1.9 (h - 60) + constant(body type)
- If we use a one-hot encoding for the body type, then together we have three models differing by the offset.
  - What is we used a different encoding?

Agenda

- Lesson
  - Features
    - Encoding text, images, recordings into numbers
  - Kernels
    - Relationships between features
  - Labels
- Demo
  - Features for Regression

Take-Aways

- What does it mean for the model to be linear?
  - Does it have to be linear in the input?
  - Does it have to be linear in the feature?
  - Does it have to be linear in the weight?

Features

- How to avoid Estimation Error with more and more features?
  - Some features are more important than other. We want to remove irrelevant features to prevent overfitting.
  - Approach
    - Select subset of features for the model
    - Score each set of features
    - Select set of features with best scores

Features

- How to avoid Estimation Error with more and more features?
  - Some features are more important than other. We want to remove irrelevant features to prevent overfitting.
  - Approach
    - Select subset of features for the model
    - Score each set of features
    - Select set of features with best scores

Example

- Forward feature selection
  - F_0 is empty.
  - For F_t choose hypothesis function h_t.
  - Select next best feature X_i.
    - Here compared to h_t
  - Set $F_{t+1} = F_t \cup X_i$
  - Repeat

Features

- How to avoid Estimation Error with more and more features?
  - Some features are more important than other. We want to remove irrelevant features to prevent overfitting.
  - Approach
    - Select subset of features for the model
    - Score each set of features
    - Select set of features with best scores

Example

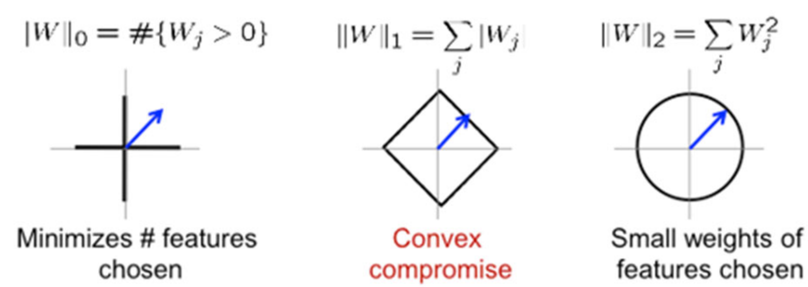
- Backward feature selection
  - F_0 contains all features.
  - For F_t choose hypothesis function h_t.
  - Select next worst feature X_i.
    - Here compared to h_t
  - Set $F_{t+1} = F_t - X_i$
  - Repeat

Features

- How to avoid Estimation Error with more and more features?
  - Some features are more important than other. We want to remove irrelevant features to prevent overfitting.
  - Approach
    - Select subset of features for the model
    - Score each set of features
    - Select set of features with best scores

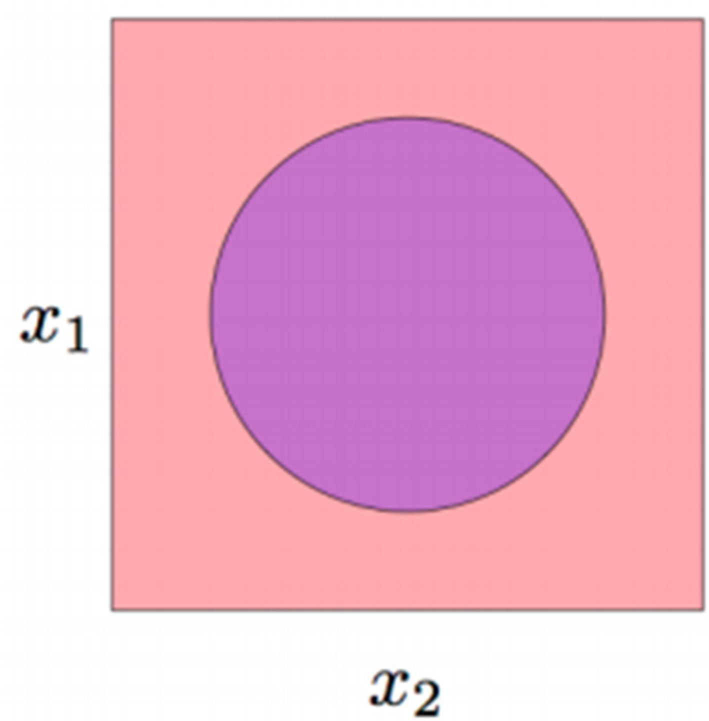
Example

- Regularization



Features

- How to avoid Optimization Error with more and more features?
  - Suppose we want to use linear classification for these categories
  - Can we use $f(x) = [x_1, x_2]$?



Features

- How to avoid Optimization Error with more and more features?
  - Suppose we have three input variables x_1, x_2, x_3
  - How many expressions can we form involving r of them?

$$\binom{r+3-1}{r}$$

1	|	3
2	|	6
3	|	10
4	|	15
5	|	21
6	|	28
7	|	36
8	|	45
9	|	55

Features

- How to avoid Optimization Error with more and more features?
  - Suppose we have three input variables x_1, x_2, x_3
  - How many expressions can we form involving r of them?

$$\binom{r + 30 - 1}{r}$$

1	|	30
2	|	465
3	|	4960
4	|	40920
5	|	278256
6	|	1623160
7	|	8347680
8	|	38608020
9	|	163011640

Features

- How to avoid Optimization Error with more and more features?
  - Suppose we want all powers up to 2 for d variables

Has dimension $O(d^2)$

$$\phi(x) = (x_1, \ldots, x_d, x_1^2, \ldots, x_d^2, \sqrt{2}x_1x_2, \ldots, \sqrt{2}x_ix_j, \ldots \sqrt{2}x_{d-1}x_d)^T$$

Features

- How to avoid Optimization Error with more and more features?
  - Suppose we want all powers up to 2 for d variables
  - The expression can be formed from inner products

Take O(d) operations for calculation

$$k(x,x') = \langle \phi(x), \phi(x') \rangle = \langle x, x' \rangle + \langle x, x' \rangle^2$$

$$\phi(x) = (x_1, \ldots, x_d, x_1^2, \ldots, x_d^2, \sqrt{2}x_1x_2, \ldots, \sqrt{2}x_ix_j, \ldots \sqrt{2}x_{d-1}x_d)^T$$

Agenda

- Lesson
  - Features
    - Encoding text, images, recordings into numbers
  - Kernels
    - Relationships between features
  - Labels
- Demo
  - Features for Regression

Objectives

- Can we extend linear regression / classification with different features?
- **Could we replace features with relationships between features in some algorithms?**
- How can we study more than two categories for classification?
- **Readings**:
  - Shalev-Schwarz Chapter 16 (see 16.3 for modified SGD)
  - Murphy Chapter 14.5 (see 14.5.2.4 for multiclass classification)

Kernels

- The dual problem for SVM just depends on the inner products of the points in the sample
- How could we change to other relationships besides the ***linear kernel***?

$$
\begin{aligned}
\sup_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j {x_j}^T x_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\
& \alpha_i \in \left[0, \frac{c}{n}\right] \; i = 1, \ldots, n.
\end{aligned}
$$

Kernels

- The dual problem for SVM just depends on the inner products of the points in the sample
- How could we change to other relationships besides the ***linear kernel***?

Gram matrix

$$K = (\langle x_i, x_j \rangle)_{i,j} = \begin{pmatrix} \langle x_1, x_1 \rangle & \cdots & \langle x_1, x_n \rangle \\ \vdots & \ddots & \cdots \\ \langle x_n, x_1 \rangle & \cdots & \langle x_n, x_n \rangle \end{pmatrix}$$

$$\sup_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j {x_j}^T x_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\alpha_i \in \left[0, \frac{c}{n}\right] \quad i = 1, \ldots, n.$$

Kernels

$K = XX^T$

- The dual problem for SVM just depends on the inner products of the points in the sample
- How could we change to other relationships besides the ***linear kernel***?

Gram matrix

$$K = (\langle x_i, x_j \rangle)_{i,j} = \begin{pmatrix} \langle x_1, x_1 \rangle & \cdots & \langle x_1, x_n \rangle \\ \vdots & \ddots & \cdots \\ \langle x_n, x_1 \rangle & \cdots & \langle x_n, x_n \rangle \end{pmatrix}$$

$$\begin{aligned} \sup_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_j^T x_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & \alpha_i \in \left[0, \frac{c}{n}\right] \quad i = 1, \ldots, n. \end{aligned}$$

Kernels

- The dual problem for SVM just depends on the inner products of the points in the sample
- How could we change to other relationships besides the ***linear kernel***?

Gram matrix

$$K = (\langle x_i, x_j \rangle)_{i,j} = \begin{pmatrix} \langle x_1, x_1 \rangle & \cdots & \langle x_1, x_n \rangle \\ \vdots & \ddots & \cdots \\ \langle x_n, x_1 \rangle & \cdots & \langle x_n, x_n \rangle \end{pmatrix}$$

$$\begin{aligned} \sup_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K_{ji} \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & \alpha_i \in \left[0, \frac{c}{n}\right] \quad i = 1, \ldots, n. \end{aligned}$$

Kernels

Since weights are combination of the points in the training set, we can even make predictions through inner products

- The dual problem for SVM just depends on the inner products of the points in the sample
- How could we change to other relationships besides the ***linear kernel***?

Gram matrix

$$K = (\langle x_i, x_j \rangle)_{i,j} = \begin{pmatrix} \langle x_1, x_1 \rangle & \cdots & \langle x_1, x_n \rangle \\ \vdots & \ddots & \cdots \\ \langle x_n, x_1 \rangle & \cdots & \langle x_n, x_n \rangle \end{pmatrix}$$

$$\begin{aligned} \sup_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K_{ji} \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & \alpha_i \in \left[0, \frac{c}{n}\right] \quad i = 1, \ldots, n. \end{aligned}$$

Kernels

- The dual problem for SVM just depends on the inner products of the points in the sample
- How could we change to other relationships besides the ***linear kernel***?

$$
\begin{aligned}
\langle w, \psi(x) \rangle &= \left\langle \sum_{i=1}^{n} \alpha_i \psi(x_i), \psi(x) \right\rangle \\
&= \sum_{i=1}^{n} \alpha_i \langle \psi(x_i), \psi(x) \rangle \\
&= \sum_{i=1}^{n} \alpha_i k(x_i, x)
\end{aligned}
$$

$$
\begin{aligned}
\sup_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K_{ji} \\
\text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\
& \alpha_i \in \left[0, \frac{c}{n}\right] \quad i = 1, \ldots, n.
\end{aligned}
$$

Kernels

Example

- Consider feature encoding for strings representing amino acids.
  - The characters are $\boxed{\{A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V\}}$

Kernels

Example

- Consider feature encoding for strings representing amino acids.
  - The characters are $\boxed{\{A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V\}}$
- How should we relate the following strings…

```
IPTSALVKETLALLSTHRTLLIANETLRIPVPVHKNHQLCTEEIFQGIGTLESQTVQGGTV
ERLFKNLSLIKKYIDGQKKKCGEERRRVNQFLDYLQEFLGVMNTEWI
```

```
PHRRDLCSRSIWLARKIRSDLTALTESYVKHQGLWSELTEAERLQENLQAYRTFHVLLA
RLLEDQQVHFTPTEGDFHQAIHTLLLQVAAFAYQIEELMILLEYKIPRNEADGMLFEKK
LWGLKVLQELSQWTVRSIHDLRFISSHQTGIP
```

Kernels

Generalizes bag-of-words encoding

Example

- Consider feature encoding for strings representing amino acids.
  - The characters are $\{A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$
- How should we relate the following strings…

IPTSALVKETLALLSTHRTLLIANETLRIPVPVHKNHQLCTEEIFQGIGTLESQTVQGGTV
ERLFKNLSLIKKYIDGQKKKCGEERRRVNQFLDY<u>LQE</u>FLGVMNTEWI

$$\kappa(x, x') = \sum_{s \in \mathcal{A}^*} w_s \phi_s(x) \phi_s(x')$$

PHRRDLCSRSIWLARKIRSDLTALTESYVKHQGLWSELTEAER<u>LQE</u>NLQAYRTFHVLLA
RLLEDQQVHFTPTEGDFHQAIHTLLLQVAAFAYQIEELMILLEYKIPRNEADGMLFEKK
LWGLKV<u>LQE</u>LSQWTVRSIHDLRFISSHQTGIP

Kernels

Use data structure called trie to efficiently compute common substrings

Example

- Consider feature encoding for strings representing amino acids.
  - The characters are $\{A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$
- How should we relate the following strings…

IPTSALVKETLALLSTHRTLLIANETLRIPVPVHKNHQLCTEEIFQGIGTLESQTVQGGTV
ERLFKNLSLIKKYIDGQKKKCGEERRRVNQFLDY<u>LQE</u>FLGVMNTEWI

$$\kappa(x, x') = \sum_{s \in \mathcal{A}^*} w_s \phi_s(x) \phi_s(x')$$

PHRRDLCSRSIWLARKIRSDLTALTESYVKHQGLWSELTEAER<u>LQE</u>NLQAYRTFHVLLA
RLLEDQQVHFTPTEGDFHQAIHTLLLQVAAFAYQIEELMILLEYKIPRNEADGMLFEKK
LWGLKV<u>LQE</u>LSQWTVRSIHDLRFISSHQTGIP

Kernels

- Can we express regression in terms of kernels?
  - We can use kernels for Ridge Regression.
  - We cannot use kernels for Lasso Regression.
- Take X = R^d and Y = R. Suppose we have n points in training set.
- The Ridge Regression objective is

$$X = \begin{pmatrix} -x_1- \\ \vdots \\ -x_n- \end{pmatrix}$$

$$J(w) = ||Xw - y||^2 + \lambda ||w||^2$$

Kernels

- Can we express regression in terms of kernels?
  - We can use kernels for Ridge Regression.
  - We cannot use kernels for Lasso Regression.
- Take X = R^d and Y = R. Suppose we have n points in training set.
- The Ridge Regression objective is

$$J(w) = ||Xw - y||^2 + \lambda||w||^2$$

$$\begin{aligned} J(w) &= (Xw - y)^T (Xw - y) + \lambda w^T w \\ \partial_w J(w) &= 2X^T (Xw - y) + 2\lambda w \\ \partial_w J(w) = 0 &\iff 2X^T X w + 2\lambda w - 2X^T y = 0 \\ &\iff (X^T X + \lambda I) w = X^T y \\ &\iff w = (X^T X + \lambda I)^{-1} X^T y \end{aligned}$$

Kernels

- Can we express regression in terms of kernels?
  - We can use kernels for Ridge Regression.
  - We cannot use kernels for Lasso Regression.
- Take X = R^d and Y = R. Suppose we have n points in training set.
- The Ridge Regression objective is

$$J(w) = ||Xw - y||^2 + \lambda||w||^2$$

$$\begin{aligned}
J(w) &= (Xw-y)^T(Xw-y) + \lambda w^T w \\
\partial_w J(w) &= 2X^T(Xw-y) + 2\lambda w \\
\partial_w J(w) = 0 &\iff 2X^T Xw + 2\lambda w - 2X^T y = 0 \\
&\iff (X^T X + \lambda I)w = X^T y \\
&\iff w = (X^T X + \lambda I)^{-1} X^T y
\end{aligned}$$

$$\begin{aligned}
w &= X^T \left[\frac{1}{\lambda}(y - Xw)\right] \\
\Longrightarrow \alpha &= \frac{1}{\lambda}(y - Xw)
\end{aligned}$$

Kernels

- Can we express regression in terms of kernels?
  - We can use kernels for Ridge Regression.
  - We cannot use kernels for Lasso Regression.
- Take X = R^d and Y = R. Suppose we have n points in training set.
- The Ridge Regression objective is

$$J(w) = ||Xw - y||^2 + \lambda ||w||^2$$

$$\begin{aligned} \alpha &= \lambda^{-1}(y - Xw) \\ \lambda\alpha &= y - XX^T\alpha \\ XX^T\alpha + \lambda\alpha &= y \\ (XX^T + \lambda I)\,\alpha &= y \\ \alpha &= (\lambda I + XX^T)^{-1} y \end{aligned}$$

$$\begin{aligned} w &= X^T\left[\frac{1}{\lambda}(y - Xw)\right] \\ \implies \alpha &= \frac{1}{\lambda}(y - Xw) \end{aligned}$$

Kernels

- How can we determine algorithms involving kernels?
  - If an algorithm minimizing an objective of a certain form, then it involves kernels

$$J(w) = R(\|w\|) + L(\langle w, \psi(x_1)\rangle, \ldots, \langle w, \psi(x_n)\rangle),$$

$\|\cdot\|$ *is the norm corresponding to the inner product* (*i.e.* $\|w\| = \sqrt{\langle w, w\rangle}$)

$R : \mathbf{R}^{\geqslant 0} \to \mathbf{R}$ *is nondecreasing* (***Regularization term***), *and*

$L : \mathbf{R}^n \to \mathbf{R}$ *is arbitrary* (***Loss term***).

Kernels

1. Let w^* be a minimizer.
2. Let $M = \text{span}\,(\psi(x_1), \ldots, \psi(x_n))$. [the "**span of the data**"]
3. Let $w = \text{Proj}_M w^*$. So $\exists \alpha$ s.t. $w = \sum_{i=1}^{n} \alpha_i \psi(x_i)$.
4. Then $w^\perp := w^* - w$ is orthogonal to M.
5. Projections decrease norms: $\|w\| \leqslant \|w^*\|$.
6. Since R is nondecreasing, $R(\|w\|) \leqslant R(\|w^*\|)$.
7. By (4), $\langle w^*, \psi(x_i) \rangle = \left\langle w + w^\perp, \psi(x_i) \right\rangle = \langle w, \psi(x_i) \rangle$.
8. $L(\langle w^*, \psi(x_1) \rangle, \ldots, \langle w^*, \psi(x_n) \rangle) = L(\langle w, \psi(x_1) \rangle, \ldots, \langle w, \psi(x_n) \rangle)$
9. $J(w) \leqslant J(w^*)$.
10. Therefore $w = \sum_{i=1}^{n} \alpha_i \psi(x_i)$ is also a minimizer.

Kernels

- When do kernels arise from inner products?
- We may not be able to determine the encoding of features for the kernel.
- However, if the Gram matrix has a certain property, then we know the existence of an encoding.

A real, symmetric matrix $M \in \mathbf{R}^{n \times n}$ is **positive semidefinite (psd)** if for any $x \in \mathbf{R}^n$,

$$x^T M x \geqslant 0.$$

Kernels

- When do kernels arise from inner products?
- We may not be able to determine the encoding of features for the kernel.
- However, if the Gram matrix has a certain property, then we know the existence of an encoding.

A symmetric function $k(x, x')$ can be expressed as an inner product

$$k(x, x') = \langle \psi(x), \psi(x') \rangle$$

for some ψ if and only if $k(x, x')$ is ***positive semidefinite.***

A real, symmetric matrix $M \in \mathbf{R}^{n \times n}$ is **positive semidefinite (psd)** if for any $x \in \mathbf{R}^n$,

$$x^T M x \geqslant 0.$$

Agenda

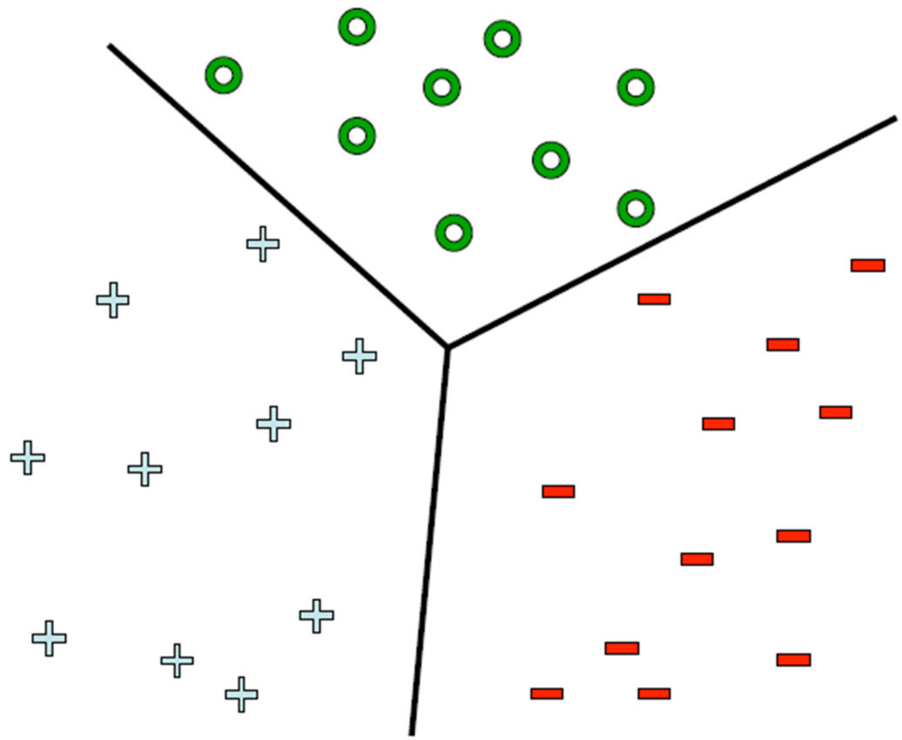
- Lesson
  - Features
    - Encoding text, images, recordings into numbers
  - Kernels
    - Relationships between features
  - Labels
- Demo
  - Features for Regression
    - Homework 3

Objectives

- Can we extend linear regression / classification with different features?
- Could we replace features with relationships between features in some algorithms?
- **How can we study more than two categories for classification?**
- **Readings:**
  - Shalev-Schwarz Chapter 16 (see 16.3 for modified SGD)
  - Murphy Chapter 14.5 (see 14.5.2.4 for multiclass classification)

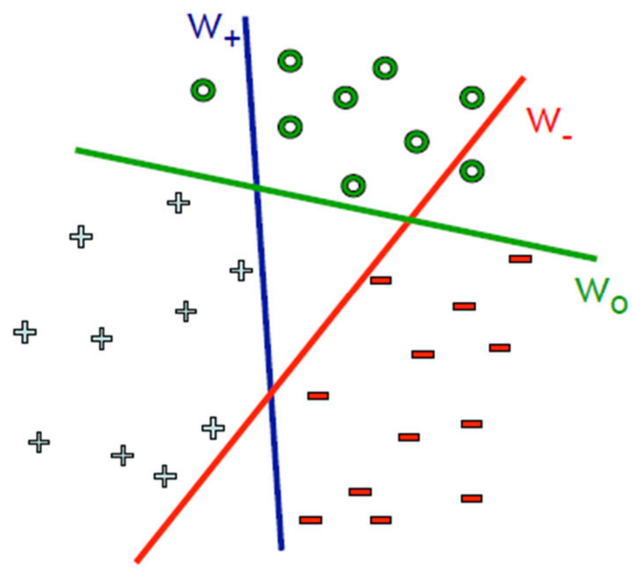
Labels

- We determine features to improve the models. Can we determine labels to improve the models?
- Multiple Categories
  - One-vs-All
  - One-vs-One



Labels

- We determine features to improve the models. Can we determine labels to improve the models?
- Multiple Categories
  - One-vs-All



Learn 3 classifiers:

- \- vs {o,+}, weights w_-
- \+ vs {o,-}, weights w_+
- o vs {+,-}, weights w_o

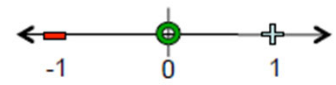
Predict label using:

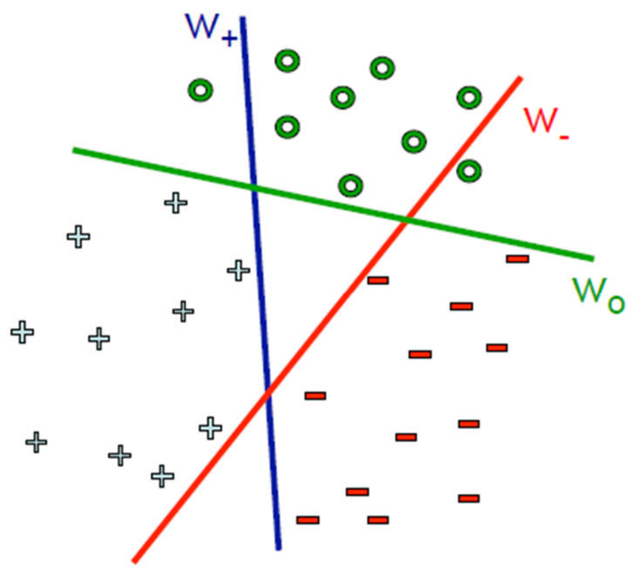
$$\hat{y} \leftarrow \arg\max_k \; w_k \cdot x + b_k$$

Labels

- We determine features to improve the models. Can we determine labels to improve the models?
- Multiple Categories
  - One-vs-All
- Issues
  - Large datasets
  - Different Scales
  - Imbalanced Data
  - Not Separable

Could we learn this (1-D) dataset? →





Learn 3 classifiers:

- \- vs {o,+}, weights w_-
- \+ vs {o,-}, weights w_+
- o vs {+,-}, weights w_o

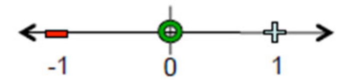
Predict label using:

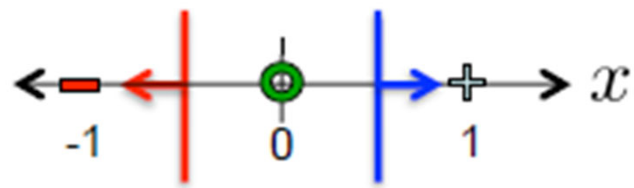
$$\hat{y} \leftarrow \arg\max_k \; w_k \cdot x + b_k$$

Labels

- We determine features to improve the models. Can we determine labels to improve the models?
- Multiple Categories
  - One-vs-All
  - One-vs-One
- Issues
  - Large datasets
  - Different Scales
  - Imbalanced Data
  - Not Separable

Could we learn this (1-D) dataset? →





$$w_- = -1 \qquad w_+ = 1$$

$$b_- = -.5 \qquad b_+ = -.5$$

$$w_o = 0$$

$$b_o = .001$$