Outline

This set of slides gives a real example of using dual problems

- Basic concepts: SVM and kernels
- SVM primal/dual problems
- Logistic Regression
- Loss Functions

Outline

- Basic concepts: SVM and kernels
- SVM primal/dual problems
- Logistic Regression
- Loss Functions

Data Classification

- Given training data in different classes (labels known)

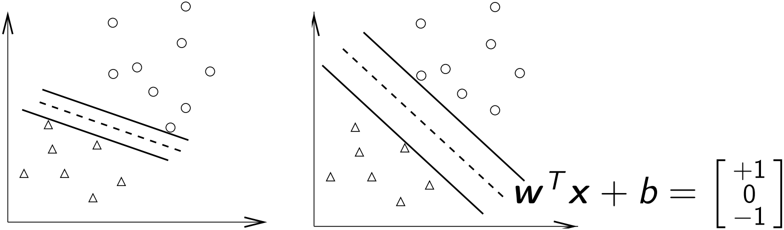
  Predict test data (labels unknown)
- Training and testing

Support Vector Classification

- Training vectors : $\boldsymbol{x}_i, i = 1, \ldots, l$
- Feature vectors. For example,
  A patient = [height, weight, ...]^T
- Consider a simple case with two classes:
  Define an indicator vector **y**

$$y_i = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \text{ in class 1} \\ -1 & \text{if } \boldsymbol{x}_i \text{ in class 2} \end{cases}$$

- A hyperplane which separates all data



- A separating hyperplane: $\boldsymbol{w}^T\boldsymbol{x} + b = 0$

$$\begin{aligned} (\boldsymbol{w}^T\boldsymbol{x}_i) + b \geq 1 & \quad \text{if } y_i = 1 \\ (\boldsymbol{w}^T\boldsymbol{x}_i) + b \leq -1 & \quad \text{if } y_i = -1 \end{aligned}$$

- Decision function $f(\boldsymbol{x}) = \text{sgn}(\boldsymbol{w}^T\boldsymbol{x} + b)$, $\boldsymbol{x}$: test data

  Many possible choices of $\boldsymbol{w}$ and b

Maximal Margin

- Distance between $\boldsymbol{w}^T\boldsymbol{x} + b = 1$ and -1:

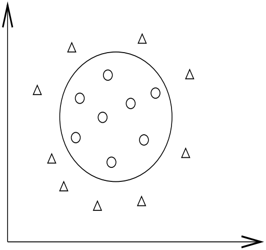
$$2/\|\boldsymbol{w}\| = 2/\sqrt{\boldsymbol{w}^T\boldsymbol{w}}$$

- A quadratic programming problem (Boser et al., 1992)

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \\ \text{subject to} \quad & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1, \\ & i = 1, \ldots, l. \end{aligned}$$

Data May Not Be Linearly Separable

- An example:



- Allow training errors
- Higher dimensional (maybe infinite) feature space

$$\phi(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots]^T.$$

- Standard SVM (Boser et al., 1992; Cortes and Vapnik, 1995)

$$\begin{aligned}\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0,\ i = 1, \ldots, l.\end{aligned}$$

- Example: $\boldsymbol{x} \in R^3, \phi(\boldsymbol{x}) \in R^{10}$

$$\begin{aligned}\phi(\boldsymbol{x}) = & [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_3, x_1^2, \\ & x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3]^T\end{aligned}$$

Finding the Decision Function

- $\boldsymbol{w}$: maybe infinite variables
- The dual problem: finite number of variables

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \boldsymbol{e}^T \boldsymbol{\alpha} \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C, i = 1, \ldots, l \\ & \mathbf{y}^T \boldsymbol{\alpha} = 0, \end{aligned}$$

where $Q_{ij} = y_i y_j \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$ and $\boldsymbol{e} = [1, \ldots, 1]^T$

- At optimum

$$\boldsymbol{w} = \sum_{i=1}^{l} \alpha_i y_i \phi(\boldsymbol{x}_i)$$

- A finite problem: #variables = #training data

Kernel Tricks

- $Q_{ij} = y_i y_j \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$ needs a closed form
- Example: $\boldsymbol{x}_i \in R^3, \phi(\boldsymbol{x}_i) \in R^{10}$

$$\phi(\boldsymbol{x}_i) = [1, \sqrt{2}(x_i)_1, \sqrt{2}(x_i)_2, \sqrt{2}(x_i)_3, (x_i)_1^2, (x_i)_2^2, (x_i)_3^2, \sqrt{2}(x_i)_1(x_i)_2, \sqrt{2}(x_i)_1(x_i)_3, \sqrt{2}(x_i)_2(x_i)_3]^T$$

Then $\phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j) = (1 + \boldsymbol{x}_i^T \boldsymbol{x}_j)^2$.

- Kernel: $K(\boldsymbol{x}, \mathbf{y}) = \phi(\boldsymbol{x})^T \phi(\mathbf{y})$; common kernels:

$$e^{-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}, \text{ (Radial Basis Function)}$$

$$(\boldsymbol{x}_i^T \boldsymbol{x}_j / a + b)^d \text{ (Polynomial kernel)}$$

Can be inner product in infinite dimensional space

Assume $x \in R^1$ and $\gamma > 0$.

$$
\begin{aligned}
& e^{-\gamma \|x_i - x_j\|^2} = e^{-\gamma (x_i - x_j)^2} = e^{-\gamma x_i^2 + 2\gamma x_i x_j - \gamma x_j^2} \\
=& e^{-\gamma x_i^2 - \gamma x_j^2} \left(1 + \frac{2\gamma x_i x_j}{1!} + \frac{(2\gamma x_i x_j)^2}{2!} + \frac{(2\gamma x_i x_j)^3}{3!} + \cdots \right) \\
=& e^{-\gamma x_i^2 - \gamma x_j^2} \left(1 \cdot 1 + \sqrt{\frac{2\gamma}{1!}} x_i \cdot \sqrt{\frac{2\gamma}{1!}} x_j + \sqrt{\frac{(2\gamma)^2}{2!}} x_i^2 \cdot \sqrt{\frac{(2\gamma)^2}{2!}} x_j^2 \right. \\
& \left. + \sqrt{\frac{(2\gamma)^3}{3!}} x_i^3 \cdot \sqrt{\frac{(2\gamma)^3}{3!}} x_j^3 + \cdots \right) = \phi(x_i)^T \phi(x_j),
\end{aligned}
$$

where

$$
\phi(x) = e^{-\gamma x^2} \left[1, \sqrt{\frac{2\gamma}{1!}} x, \sqrt{\frac{(2\gamma)^2}{2!}} x^2, \sqrt{\frac{(2\gamma)^3}{3!}} x^3, \cdots \right]^T.
$$

Decision function

- At optimum

$$\boldsymbol{w} = \sum_{i=1}^{l} \alpha_i y_i \phi(\boldsymbol{x}_i)$$

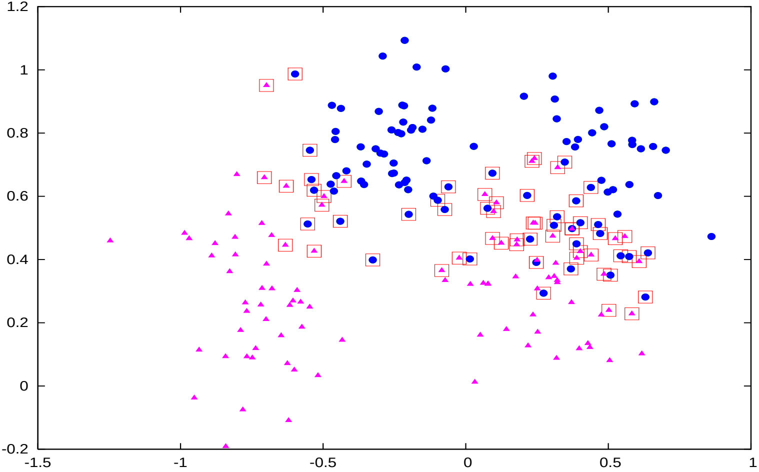
- Decision function

$$\begin{aligned}
& \boldsymbol{w}^T \phi(\boldsymbol{x}) + b \\
= & \sum_{i=1}^{l} \alpha_i y_i \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}) + b \\
= & \sum_{i=1}^{l} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b
\end{aligned}$$

- Only $\phi(\boldsymbol{x}_i)$ of $\alpha_i > 0$ used $\Rightarrow$ support vectors

Support Vectors: More Important Data

Only $\phi(\boldsymbol{x}_i)$ of $\alpha_i > 0$ used $\Rightarrow$ support vectors



Outline

- Basic concepts: SVM and kernels
- SVM primal/dual problems
- Logistic Regression
- Loss Functions

Deriving the Dual

- Consider the problem without ξ_i

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \\ \text{subject to} \quad & y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i)+b) \geq 1, i = 1,\ldots,l. \end{aligned}$$

- Its dual

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \boldsymbol{e}^T\boldsymbol{\alpha} \\ \text{subject to} \quad & 0 \leq \alpha_i, \qquad i = 1,\ldots,l, \\ & \mathbf{y}^T\boldsymbol{\alpha} = 0. \end{aligned}$$

Lagrangian Dual

$$\max_{\boldsymbol{\alpha} \geq 0} \left(\min_{\boldsymbol{w}, b} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) \right),$$

where

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{l} \alpha_i \left(y_i(\boldsymbol{w}^T \phi(\boldsymbol{x}_i) + b) - 1 \right)$$

Strong duality

$$\min \text{ Primal} = \max_{\boldsymbol{\alpha} \geq 0} \left(\min_{\boldsymbol{w}, b} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) \right)$$

- Simplify the dual. When $\boldsymbol{\alpha}$ is fixed,

$$\min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) =$$

$$\begin{cases} -\infty & \text{if } \sum_{i=1}^{l} \alpha_i y_i \neq 0 \\ \min_{\boldsymbol{w}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_{i=1}^{l} \alpha_i [y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - 1] & \text{if } \sum_{i=1}^{l} \alpha_i y_i = 0 \end{cases}$$

- If $\sum_{i=1}^{l} \alpha_i y_i \neq 0$,
  decrease

$$-b \sum_{i=1}^{l} \alpha_i y_i$$

in $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ to $-\infty$

- If $\sum_{i=1}^{l} \alpha_i y_i = 0$, optimum of the strictly convex $\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_{i=1}^{l} \alpha_i [y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - 1]$ happens when

$$\frac{\partial}{\partial \boldsymbol{w}} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = 0.$$

- Thus,

$$\boldsymbol{w} = \sum_{i=1}^{l} \alpha_i y_i \phi(\boldsymbol{x}_i).$$

- Note that

$$\begin{aligned} \boldsymbol{w}^T\boldsymbol{w} &= \left(\sum_{i=1}^{l}\alpha_i y_i \phi(\boldsymbol{x}_i)\right)^T \left(\sum_{j=1}^{l}\alpha_j y_j \phi(\boldsymbol{x}_j)\right) \\ &= \sum_{i,j}\alpha_i\alpha_j y_i y_j \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j) \end{aligned}$$

- The dual is

$$\max_{\boldsymbol{\alpha}\geq 0} \begin{cases} \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j) & \text{if } \sum_{i=1}^{l}\alpha_i y_i = 0 \\ -\infty & \text{if } \sum_{i=1}^{l}\alpha_i y_i \neq 0 \end{cases}$$

- Lagrangian dual: $\max_{\boldsymbol{\alpha} \geq 0}(\min_{\boldsymbol{w}, b} L(\boldsymbol{w}, b, \boldsymbol{\alpha}))$
- $-\infty$ definitely not maximum of the dual

  Dual optimal solution not happen when

$$\sum_{i=1}^{l} \alpha_i y_i \neq 0$$

  .
- Dual simplified to

$$\begin{aligned} \max_{\boldsymbol{\alpha} \in R^l} \quad & \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j) \\ \text{subject to} \quad & \mathbf{y}^T \boldsymbol{\alpha} = 0, \\ & \alpha_i \geq 0, i = 1, \ldots, l. \end{aligned}$$

- Our problems may be infinite dimensional
- Can still use Lagrangian duality

  See a rigorous discussion in Lin (2001)

Outline

- Basic concepts: SVM and kernels
- SVM primal/dual problems
- Logistic Regression
- Loss Functions

Logistic Regression

- For a label-feature pair $(\boldsymbol{y},\boldsymbol{x})$, assume the probability model

$$p(y|\boldsymbol{x}) = \frac{1}{1 + e^{-y\boldsymbol{w}^T\boldsymbol{x}}}.$$

- $\boldsymbol{w}$ is the parameter to be decided
- Assume

$$(y_i, \boldsymbol{x}_i), i = 1, \ldots, l$$

  are training instances

Logistic Regression (Cont'd)

- Logistic regression finds $\boldsymbol{w}$ by maximizing the following likelihood

$$\max_{\boldsymbol{w}} \quad \prod_{i=1}^{l} p\left(y_i | \boldsymbol{x}_i\right). \tag{1}$$

- Regularized logistic regression

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l} \log\left(1 + e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i}\right). \tag{2}$$

C: regularization parameter decided by users

Outline

- Basic concepts: SVM and kernels
- SVM primal/dual problems
- Logistic Regression
- Loss Functions

- We derive SVM from the viewpoint of maximal margin
- We derive logistic regression from minimizing the negative log likelihood
- They can both be considered from the viewpoint of regularized linear classification

Minimizing Training Errors

- Basically a classification method starts with minimizing the training errors

$$\min_{\text{model}} \quad \text{(training errors)}$$

- That is, all or most training data with labels should be correctly classified by our model
- A model can be a decision tree, a support vector machine, a neural networks, or other types

Minimizing Training Errors (Cont'd)

- We consider the model to be a vector $\boldsymbol{w}$
- That is, the decision function is

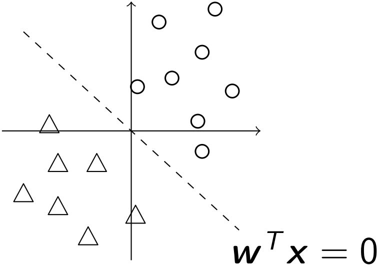
$$\text{sgn}(\boldsymbol{w}^T\boldsymbol{x})$$

- For any data, $\boldsymbol{x}$, the predicted label is

$$\begin{cases} 1 & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Minimizing Training Errors (Cont'd)

- The two-dimensional situation



- This seems to be quite restricted, but practically $\boldsymbol{x}$ is in a much higher dimensional space

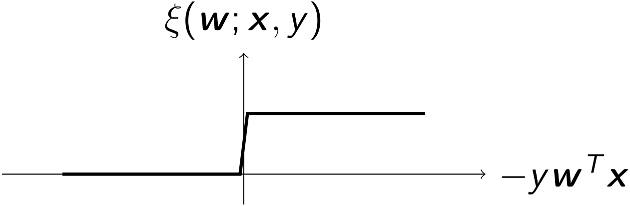
Minimizing Training Errors (Cont'd)

- To characterize the training error, we need a loss function $\xi(\boldsymbol{w}; \boldsymbol{x}, y)$ for each instance $(\boldsymbol{x}, y)$
- Ideally we should use 0–1 training loss:

$$\xi(\boldsymbol{w}; \boldsymbol{x}, y) = \begin{cases} 1 & \text{if } y\boldsymbol{w}^T\boldsymbol{x} < 0, \\ 0 & \text{otherwise} \end{cases}$$

Minimizing Training Errors (Cont'd)

- However, this function is discontinuous. The optimization problem becomes difficult



- We need continuous approximations

Loss Functions

- Some commonly used ones:

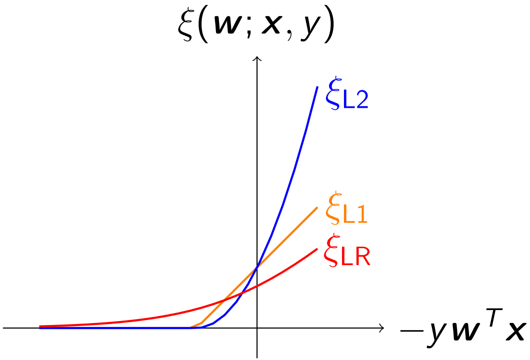
$$\xi_{\text{L1}}(\boldsymbol{w};\boldsymbol{x},y) \equiv \max(0, 1 - y\boldsymbol{w}^T\boldsymbol{x}), \tag{3}$$

$$\xi_{\text{L2}}(\boldsymbol{w};\boldsymbol{x},y) \equiv \max(0, 1 - y\boldsymbol{w}^T\boldsymbol{x})^2, \tag{4}$$

$$\xi_{\text{LR}}(\boldsymbol{w};\boldsymbol{x},y) \equiv \log(1 + e^{-y\boldsymbol{w}^T\boldsymbol{x}}). \tag{5}$$

- SVM (Boser et al., 1992; Cortes and Vapnik, 1995): (3)-(4)
- Logistic regression (LR): (5)

Loss Functions (Cont'd)



Their performance is usually similar

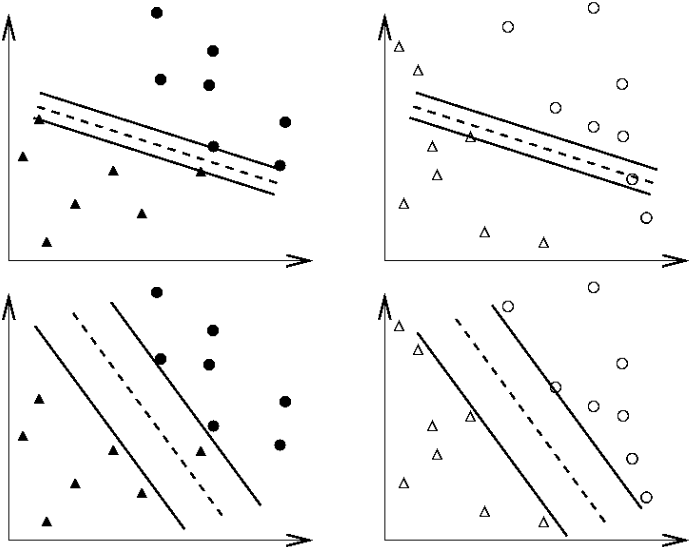
Common Loss Functions (Cont'd)

- However, minimizing training losses may not give a good model for future prediction
- Overfitting occurs

Overfitting

- See the illustration in the next slide
- For classification,
  You can easily achieve 100% training accuracy
- This is useless
- When training a data set, we should
  Avoid underfitting: small training error
  Avoid overfitting: small testing error

● and ▲: training; ○ and △: testing



Regularization

- To minimize the training error we manipulate the $\boldsymbol{w}$ vector so that it fits the data
- To avoid overfitting we need a way to make $\boldsymbol{w}$'s values less extreme.
- One idea is to make $\boldsymbol{w}$ values closer to zero
- We can add, for example,

$$\frac{\boldsymbol{w}^T\boldsymbol{w}}{2} \quad \text{or} \quad \|\boldsymbol{w}\|_1$$

to the function that is minimized

Regularized Linear Classification

- Training data $\{y_i, \boldsymbol{x}_i\}, \boldsymbol{x}_i \in R^n, i = 1, \ldots, l, \ y_i = \pm 1$
- l: # of data, n: # of features

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}), \quad f(\boldsymbol{w}) \equiv \frac{\boldsymbol{w}^T \boldsymbol{w}}{2} + C \sum_{i=1}^{l} \xi(\boldsymbol{w}; \boldsymbol{x}_i, y_i)$$

- $\boldsymbol{w}^T \boldsymbol{w}/2$: regularization term (we have no time to talk about L1 regularization here)
- $\xi(\boldsymbol{w}; \boldsymbol{x}, y)$: loss function: we hope $y \boldsymbol{w}^T \boldsymbol{x} > 0$
- C: regularization parameter

Discussion

- You can use $\|\boldsymbol{w}\|_1$ regularization. This is now popular because of sparsity (i.e., some $\boldsymbol{w}$'s components are zeros

  But do we still have maximal margin interpretation?
- For SVM, can we have an interpretation like maximum likelihood of logistic regression?
- For regularized logistic regression, can we have an interpretation of maximal margin?

References I

B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.

C. Cortes and V. Vapnik. Support-vector network. *Machine Learning*, 20:273–297, 1995.

C.-J. Lin. Formulations of support vector machines: a note from an optimization point of view. *Neural Computation*, 13(2):307–317, 2001.