STA141C: Big Data & High Performance Statistical Computing

Lecture 9: Classification

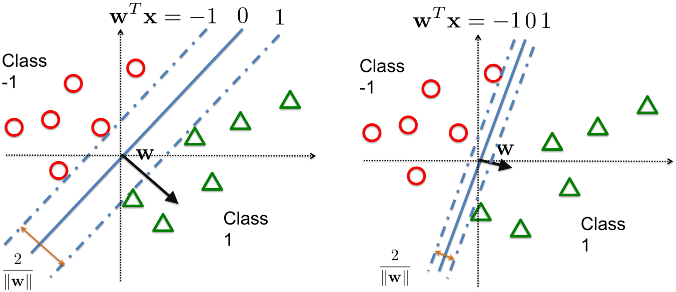
Cho-Jui Hsieh
UC Davis

May 18, 2017

Linear SVM

Support Vector Machines

- SVM is a widely used classifier.
- Given:
  - Training data points $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$.
  - Each $\boldsymbol{x}_i \in \mathbb{R}^d$ is a feature vector:
  - Consider a simple case with two classes: $y_i \in \{+1, -1\}$.
- Goal: Find a hyperplane to separate these two classes of data: if $y_i = 1$, $\boldsymbol{w}^T \boldsymbol{x}_i \geq 1$; if $y_i = -1$, $\boldsymbol{w}^T \boldsymbol{x}_i \leq -1$.



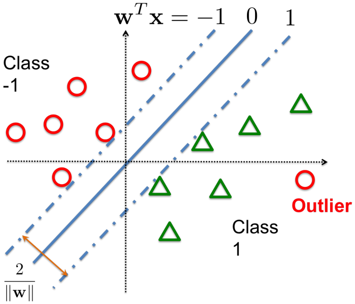
Support Vector Machines (hard constraints)

- Given training data $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathbb{R}^d$ with labels $y_i \in \{+1, -1\}$.
- SVM primal problem (with hard constraints):

$$\min_{\boldsymbol{w},\xi} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$$

$$\text{s.t. } y_i(\boldsymbol{w}^T\boldsymbol{x}_i) \geq 1, i = 1, \dots, n,$$

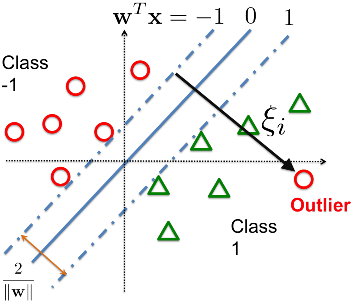
- What if there are outliers?



Support Vector Machines

- Given training data $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathbb{R}^d$ with labels $y_i \in \{+1, -1\}$.
- SVM primal problem:

$$
\begin{aligned}
\min_{\boldsymbol{w},\xi} \ & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{n}\xi_i \\
\text{s.t. } & y_i(\boldsymbol{w}^T\boldsymbol{x}_i) \geq 1 - \xi_i, i = 1, \ldots, n, \\
& \xi_i \geq 0
\end{aligned}
$$



Support Vector Machines

- SVM primal problem can be written as

$$\min_{\boldsymbol{w}} \quad \underbrace{\frac{1}{2}\|\boldsymbol{w}\|^2}_{\text{L2 regularization}} + \sum_{i=1}^{n} \underbrace{\max(0, 1 - y_i \boldsymbol{w}^T \boldsymbol{x}_i)}_{\text{hinge loss}}$$

- Non-differentiable when $y_i \boldsymbol{w}^T \boldsymbol{x}_i = 1$ for some i

General Empirical Risk Minimization

- Regularized ERM:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} P(\boldsymbol{w}) := \sum_{i=1}^{n} \ell_i(\boldsymbol{w}^T \boldsymbol{x}_i) + R(\boldsymbol{w})$$

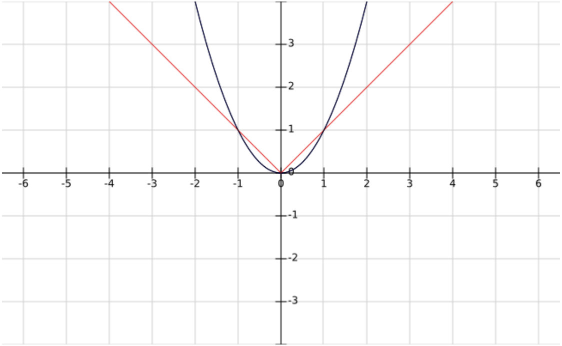
  - $\ell_i(\cdot)$: loss function
  - $R(\boldsymbol{w})$: regularization
- Dual problem may have a different form (?)

Examples

- Loss functions:
  - Regression: $\ell_i(\boldsymbol{x}_i) = (\boldsymbol{x}_i - y_i)^2$
  - SVM (hinge loss): $\ell_i(\boldsymbol{x}_i) = \max(1 - y_i \boldsymbol{w}^T \boldsymbol{x}_i, 0)$
  - Square hinge loss: $\ell_i(\boldsymbol{x}_i) = \max(1 - y_i \boldsymbol{w}^T \boldsymbol{x}_i, 0)^2$
  - Logistic regression: $\ell_i(\boldsymbol{x}_i) = \log(1 + e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i})$

Examples

- Regularizations:
  - L2-regularization: $\|\boldsymbol{w}\|_2^2$: small but dense solution
  - L1-regularization: $\|\boldsymbol{w}\|_1$: sparse solution
  - Nuclear norm: $\|W\|_*$: low-rank solution



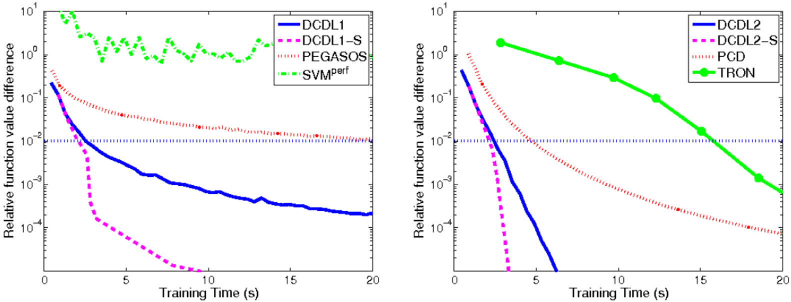
LIBLINEAR

- Implemented in LIBLINEAR:

  `https://www.csie.ntu.edu.tw/~cjlin/liblinear/`
- Other functionalities:
  - Logistic regression (L1 or L2 regularization)
  - Multi-class SVM
  - Support vector regression
  - Cross-validation

LIBLINEAR

- RCV1: 677,399 training samples; 47,236 features; 49,556,258 nonzeroes in the whole dataset.

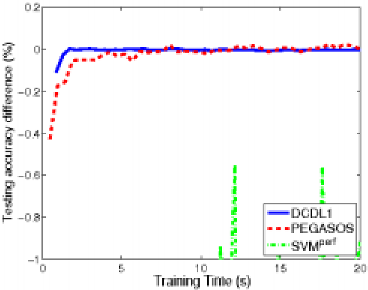


(e) L1-SVM: rcv1 (f) L2-SVM: rcv1

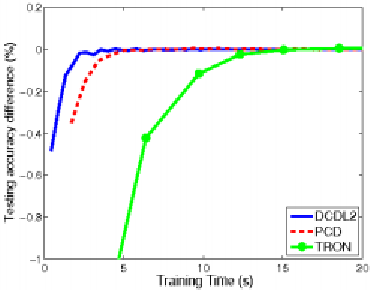
Time vs primal objective function value

LIBLINEAR

- RCV1: 677,399 training samples; 47,236 features; 49,556,258 nonzeroes in the whole dataset.



(e) L1-SVM: rcv1



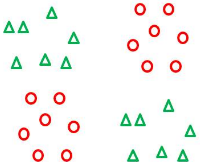
(f) L2-SVM: rcv1

Time vs prediction accuracy

Kernel SVM

Non-linearly separable problems

- What if the data is not linearly separable?



$$x \to \varphi(x) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}$$

Solution: map data $\boldsymbol{x}_i$ to higher dimensional(maybe infinite) feature space $\varphi(x_i)$, where they are linearly separable.

Support Vector Machines (SVM)

- SVM primal problem:

$$\min_{\boldsymbol{w},\xi} \ \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{n}\xi_i$$
$$\text{s.t. } \ y_i(\boldsymbol{w}^T\varphi(\boldsymbol{x}_i)) \geq 1-\xi_i, \quad \xi_i \geq 0, \ i = 1,\ldots,n,$$

- The dual problem for SVM:

$$\min_{\boldsymbol{\alpha}} \ \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \boldsymbol{e}^T\boldsymbol{\alpha},$$
$$\text{s.t. } \ 0 \leq \alpha_i \leq C, \quad \text{for } \ i = 1,\ldots,n,$$

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

- Kernel trick: define $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^T\varphi(\boldsymbol{x}_j)$.
- At optimum: $\boldsymbol{w} = \sum_{i=1}^{n}\alpha_i y_i\varphi(\boldsymbol{x}_i)$,

Various types of kernels

- Gaussian kernel: $K(\boldsymbol{x}_i, \mathbf{y}_j) = e^{-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}$;
- Polynomial kernel: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\gamma \boldsymbol{x}_i^T \boldsymbol{x}_j + c)^d$.
- Hard to solve: need to solve n-by-n quadratic minimization problem, $\geq O(n^2)$ time.
- LIBSVM: `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`
- For linear SVM, use LIBLINEAR instead of LIBSVM.

Scikit-learn

- Linear SVM: sklearn.svm.LinearSVC
- Logistic Regression: sklearn.linear_model.LogisticRegression
- Kernel SVM: sklearn.svm.SVC
- ...

Scikit-learn

- Linear SVM: sklearn.svm.LinearSVC
- Logistic Regression: sklearn.linear_model.LogisticRegression
- Kernel SVM: sklearn.svm.SVC
- ...
- Practice in homework.

Coming up

- Classification

Questions?