Linear SVM
Support Vector Machines

- SVM is a widely used classifier.
- Given:
  - Training data points $x_1, \ldots, x_n$.
  - Each $x_i \in \mathbb{R}^d$ 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 $y_i = 1$, $w^T x_i \geq 1$; if $y_i = -1$, $w^T x_i \leq -1$. 

\[ w^T x = -1 \begin{array}{c} 0 \end{array} \leq 1 \]

\[ w^T x = -1 \begin{array}{c} 0 \end{array} \leq 1 \]

\[ \frac{2}{||w||} \]

\[ \frac{2}{||w||} \]

\[ w \]

\[ w \]

\[ Class \]

- Class

- 1

- -1

- Class

- 1

- -1
Support Vector Machines (hard constraints)

- Given training data $x_1, \cdots, x_n \in \mathbb{R}^d$ with labels $y_i \in \{+1, -1\}$.
- SVM primal problem (with hard constraints):
  
  $$\min_{w, \xi} \frac{1}{2}w^T w$$

  $$\text{s.t. } y_i(w^T x_i) \geq 1, i = 1, \ldots, n,$$

- What if there are outliers?
Support Vector Machines

- Given training data $x_1, \cdots, x_n \in \mathbb{R}^d$ with labels $y_i \in \{+1, -1\}$.
- SVM primal problem:

$$\min_{\mathbf{w}, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^{n} \xi_i$$

s.t. $y_i (\mathbf{w}^T \mathbf{x}_i) \geq 1 - \xi_i$, $i = 1, \ldots, n$,

$\xi_i \geq 0$
SVM primal problem can be written as

$$\min_w \frac{1}{2} \|w\|^2 + \sum_{i=1}^{n} \max(0, 1 - y_i w^T x_i)$$

- L2 regularization
- Hinge loss

Non-differentiable when $y_i w^T x_i = 1$ for some $i$
General Empirical Risk Minimization

- Regularized ERM:

\[
\min_{w \in \mathbb{R}^d} P(w) := \sum_{i=1}^{n} \ell_i(w^T x_i) + R(w)
\]

- \(\ell_i(\cdot)\): loss function
- \(R(w)\): regularization

- Dual problem may have a different form (?)
Examples

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

- Regularizations:
  - L2-regularization: $\|w\|_2^2$: small but dense solution
  - L1-regularization: $\|w\|_1$: 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
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
RCV1: 677,399 training samples; 47,236 features; 49,556,258 nonzeros 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?

Solution: map data \( x_i \) to higher dimensional (maybe infinite) feature space \( \varphi(x_i) \), where they are linearly separable.

\[
x \rightarrow \varphi(x) = \begin{bmatrix}
x_1^2 \\
\sqrt{2}x_1x_2 \\
x_2^2
\end{bmatrix}
\]
Support Vector Machines (SVM)

- SVM primal problem:

$$\min_{w, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$

\[\text{s.t. } y_i(w^T \varphi(x_i)) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n,\]

- The dual problem for SVM:

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha,$$

\[\text{s.t. } 0 \leq \alpha_i \leq C, \quad \text{for } i = 1, \ldots, n,\]

where $Q_{ij} = y_i y_j \varphi(x_i)^T \varphi(x_j)$ and $e = [1, \ldots, 1]^T$.

- Kernel trick: define $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$.

- At optimum: $w = \sum_{i=1}^{n} \alpha_i y_i \varphi(x_i)$,
Various types of kernels

- Gaussian kernel: \( K(x_i, y_j) = e^{-\gamma \|x_i - x_j\|^2_2} \);
- Polynomial kernel: \( K(x_i, x_j) = (\gamma x_i^T x_j + c)^d \).
- 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
- ...
Linear SVM: sklearn.svm.LinearSVC
Logistic Regression: sklearn.linear_model.LogisticRegression
Kernel SVM: sklearn.svm.SVC

... 

Practice in homework.
Coming up

- Classification

Questions?