Kernel Method: Data Analysis with Positive Definite Kernels

4. Support Vector Machine

Kenji Fukumizu

The Institute of Statistical Mathematics.
Graduate University of Advanced Studies /
Tokyo Institute of Technology

Nov. 17-26, 2010
Intensive Course at Tokyo Institute of Technology

Outline

A quick course on convex optimization
- Convexity and convex optimization
- Dual problem for optimization

Optimization in learning of SVM
- Dual problem and support vectors
- Sequential Minimal Optimization (SMO)
- Other approaches

Extension of SVM
- Multiclass classification with SVM
- Combination of binary classifiers
- Structured output and others

Optimization of SVM

$$\min_{w_i,b,\xi_i} \frac{1}{2}\sum_{i,j=1}^{N} w_i w_j k(X_i, X_j) + C\sum_{i=1}^{N}\xi_i,$$

$$\text{subj. to} \quad \begin{cases} Y_i(\sum_{j=1}^{N} k(X_i, X_j) w_j + b) \geq 1 - \xi_i, \\ \xi_i \geq 0. \end{cases}$$

Quadratic programming (QP). Special case of convex optimization.

The QP for SVM can be solved in the above form, but the dual form is easier.

Convexity I

For the details on convex optimization, see [BV04].

- Convex set:

  A set C in a vector space is convex if for every $x, y \in C$ and $t \in [0, 1]$

  $$tx + (1-t)y \in C.$$

- Convex function:

  Let C be a convex set. $f : C \to \mathbb{R}$ is called a convex function if for every $x, y \in C$ and $t \in [0, 1]$

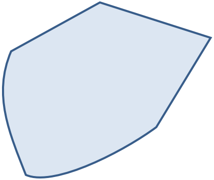
  $$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y).$$

- Concave function:

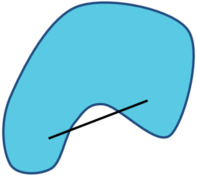
  Let C be a convex set. $f : C \to \mathbb{R}$ is called a concave function if for every $x, y \in C$ and $t \in [0, 1]$

  $$f(tx + (1-t)y) \geq tf(x) + (1-t)f(y).$$

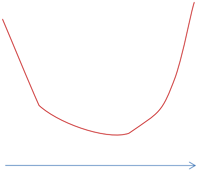
Convexity II



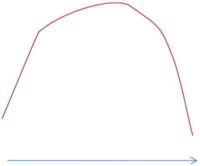
convex set



non-convex set



convex function



concave function

Convexity III

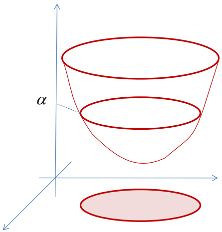
- Fact: If $f : C \to \mathbb{R}$ is a convex function, the set

$$\{x \in C \mid f(x) \leq \alpha\}$$

  is a convex set for every $\alpha \in \mathbb{R}$.
- If $f_t(x) : C \to \mathbb{R}$ ($t \in T$) are convex, then

$$f(x) = \sup_{t \in T} f_t(x)$$

  is also convex.



Convex Optimization I

- A general form of convex optimization
  $\mathcal{D}$: convex set in $\mathbb{R}^n$. $f(x)$, $h_i(x)$ $(1 \leq i \leq \ell)$: $\mathcal{D} \to \mathbb{R}$, convex functions on $\mathcal{D}$. $a_i \in \mathbb{R}^n, b_j \in \mathbb{R}$ $(1 \leq j \leq m)$.

$$\min_{x \in \mathcal{D}} f(x) \qquad \text{subject to } \begin{cases} h_i(x) \leq 0 & (1 \leq i \leq \ell), \\ a_j^T x + b_j = 0 & (1 \leq j \leq m). \end{cases}$$

  h_i: inequality constraints,
  $r_j(x) = a_j^T x + b_j$: linear equality constraints.

- Feasible set:

$$\mathcal{F} = \{x \in \mathcal{D} \mid h_i(x) \leq 0 \ (1 \leq i \leq \ell), r_j(x) = 0 \ (1 \leq j \leq m)\}.$$

  The above optimization problem is called feasible if $\mathcal{F} \neq \emptyset$.

Convex Optimization II

- Fact 1. The feasible set is a convex set.
- Fact 2. The set of minimizers

$$X_{opt} = \{x \in \mathcal{F} \mid f(x) = \inf\{f(y) \mid y \in \mathcal{F}\}\}$$

is convex. No local minima for convex optimization.

proof. The intersection of convex sets is convex, which leads (1).

Let

$$p^* = \inf_{x \in \mathcal{F}} f(x).$$

Then,

$$X_{opt} = \{x \in \mathcal{D} \mid f(x) \le p^*\} \cap \mathcal{F}.$$

Both sets in r.h.s. are convex. This proves (2) $\square$

Examples

- Linear program (LP)

$$\min c^T x \qquad \text{subject to } \begin{cases} Ax = b, \\ Gx \preceq h.^1 \end{cases}$$

The objective function, the equality and inequality constraints are all linear.

- Quadratic program (QP)

$$\min \frac{1}{2} x^T P x + q^t x + r \qquad \text{subject to } \begin{cases} Ax = b, \\ Gx \preceq h, \end{cases}$$

where P is a positive semidefinite matrix.
Objective function: quadratic.
Equality, inequality constraints: linear.

[1] $Gx \preceq h$ denotes $g_j^T x \leq h_j$ for all j, where $G = (g_1, \ldots, g_m)^T$.

Lagrange Dual

- Consider an optimization problem (which may not be convex):

$$\text{(primal)} \qquad \min_{x\in\mathcal{D}} f(x) \qquad \text{subject to } \begin{cases} h_i(x) \le 0 & (1 \le i \le \ell), \\ r_j(x) = 0 & (1 \le j \le m). \end{cases}$$

- Lagrange dual function: $g : \mathbb{R}^\ell \times \mathbb{R}^m \to [-\infty, \infty)$

$$g(\lambda, \nu) = \inf_{x\in\mathcal{D}} L(x, \lambda, \nu),$$

  where

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{\ell} \lambda_i h_i(x) + \sum_{j=1}^{m} \nu_j r_j(x).$$

  λ_i and ν_j are called Lagrange multipliers.
- g is a concave function.

Dual Problem and Weak Duality I

- Dual problem

$$\text{(dual)} \qquad \max g(\lambda,\nu) \qquad \text{subject to} \quad \lambda \succeq 0.$$

- The dual and primal problems have close connection.

Theorem 1 (weak duality)

Let

$$p^* = \inf\{f(x) \mid h_i(x) \le 0 \ (1 \le i \le \ell), r_j(x) = 0 \ (1 \le j \le m)\}.$$

and

$$d^* = \sup\{g(\lambda,\nu) \mid \lambda \succeq 0, \nu \in \mathbb{R}^m\}.$$

Then,

$$d^* \le p^*.$$

The weak duality does not require the convexity of the primal optimization problem.

Dual Problem and Weak Duality II

Proof. Let $\forall \lambda \succeq 0, \nu \in \mathbb{R}^m$.
For any feasible point x,

$$L(x, \lambda, \nu) = f(x) + \textstyle\sum_{i=1}^{\ell} \lambda_i h_i(x) + \sum_{j=1}^{m} \nu_j r_j(x) \ \leq \ f(x).$$

(The second term is non-positive, and the third term is zero.)

By taking infimum,

$$\inf_{x:feasible} L(x, \lambda, \nu) \leq p^*.$$

Thus,

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) \leq \inf_{x:feasible} L(x, \lambda, \nu) \leq p^*$$

for any $\lambda \succeq 0, \nu \in \mathbb{R}^m$. $\square$

Strong Duality

We need some conditions to obtain the strong duality $d^* = p^*$.

- Convexity of the problem: f and h_i are convex, r_j are linear.
- Slater's condition

  There is $\tilde{x} \in \mathrm{relint}\mathcal{D}$ such that

$$h_i(\tilde{x}) < 0 \quad (1 \leq \forall i \leq \ell), \qquad r_j(\tilde{x}) = a_j^T \tilde{x} + b_j = 0 \quad (1 \leq \forall j \leq m).$$

Theorem 2 (Strong duality)

Suppose the primal problem is convex, and Slater's condition holds. Then, there is $\lambda^* \geq 0$ and $\nu^* \in \mathbb{R}^m$ such that

$$g(\lambda^*, \nu^*) = d^* = p^*.$$

Proof is omitted (see [BV04] Sec.5.3.2.).

There are also other conditions to guarantee the strong duality.

Complementary Slackness I

- Consider the (not necessarily convex) optimization problem:

$$\min f(x) \qquad \text{subject to } \begin{cases} h_i(x) \le 0 & (1 \le i \le \ell), \\ r_j(x) = 0 & (1 \le j \le m). \end{cases}$$

- Assumption: the optimum of the primal/dual problems are given by x^* and (λ^*, ν^*) ($\lambda^* \succeq 0$), and they satisfy the strong duality:

$$g(\lambda^*, \nu^*) = f(x^*).$$

Complementary Slackness II

- Observation:

$$\begin{aligned} f(x^*) = g(\lambda^*, \nu^*) &= \inf_{x \in \mathcal{D}} L(x, \lambda^*, \nu^*) \qquad \text{[definition]} \\ &\leq L(x^*, \lambda^*, \nu^*) \\ &= f(x^*) + \textstyle\sum_{i=1}^{\ell} \lambda_i^* h_i(x^*) + \sum_{j=1}^{m} \nu_j^* r_j(x^*) \\ &\leq f(x^*) \qquad \text{[2nd} \leq 0 \text{ and 3rd} = 0\text{]} \end{aligned}$$

The two inequalities are in fact equalities.

Complementary Slackness III

- Consequence 1:

$$x^* \text{ minimizes } L(x, \lambda^*, \nu^*)$$

(Primal solution by unconstrained optimization)

- Consequence 2:

$$\lambda_i^* h_i(x^*) = 0 \qquad \text{for all } i$$

The latter is called complementary slackness.

Equivalently,

$$\lambda_i^* > 0 \quad \Rightarrow \quad h_i(x^*) = 0,$$

or

$$h_i(x^*) < 0 \quad \Rightarrow \quad \lambda_i^* = 0.$$

KKT Condition I

KKT conditions give useful relations between the primal and dual solutions.

- Consider the convex optimization problem.
  Assume $\mathcal{D}$ is open and $f(x)$, $h_i(x)$ are differentiable.

$$\min f(x) \qquad \text{subject to} \begin{cases} h_i(x) \leq 0 & (1 \leq i \leq \ell), \\ r_j(x) = 0 & (1 \leq j \leq m). \end{cases}$$

- x^* and (λ^*, ν^*): any optimal points of the primal and dual problems.
- Assume strong duality: $f(x^*) = g(\lambda^*, \nu^*)$.
- From Consequence 1 ($x^* = \arg\min L(x, \lambda^*, \nu^*)$),

$$\nabla f(x^*) + \textstyle\sum_{i=1}^{\ell} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{m} \nu_j^* \nabla r_j(x^*) = 0.$$

KKT Condition II

The following are necessary conditions.

Karush-Kuhn-Tucker (KKT) conditions:

$h_i(x^*) \le 0 \quad (i = 1, \dots, \ell)$ [primal constraints]

$r_j(x^*) = 0 \quad (j = 1, \dots, m)$ [primal constraints]

$\lambda_i^* \ge 0 \quad (i = 1, \dots, \ell)$ [dual constraints]

$\lambda_i^* h_i(x^*) = 0 \quad (i = 1, \dots, \ell)$ [complementary slackness]

$\nabla f(x^*) + \sum_{i=1}^{\ell} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{m} \nu_j^* \nabla r_j(x^*) = 0.$

Theorem 3 (KKT condition)

For a convex optimization problem with differentiable functions, x^* and (λ^*, ν^*) are the primal-dual solutions with strong duality if and only if they satisfy KKT conditions.

Example

- Quadratic minimization under equality constraints.

$$\min \frac{1}{2}x^T P x + q^T x + r \qquad \text{subject to} \quad Ax = b.$$

- KKT conditions:

$$\begin{aligned} &Ax^* = b, && \text{[primal constraint]} \\ &\nabla_x L(x^*, \nu^*) = 0 \quad \Longrightarrow && Px^* + q + A^T\nu^* = 0 \end{aligned}$$

- The solution is given by

$$\begin{pmatrix} P & A^T \\ A & O \end{pmatrix} \begin{pmatrix} x^* \\ \nu^* \end{pmatrix} = \begin{pmatrix} -q \\ b \end{pmatrix}.$$

Primal Problem of SVM

SVM primal problem:

$$\min_{w_i, b, \xi_i} \frac{1}{2} \sum_{i,j=1}^{N} w_i w_j k(X_i, X_j) + C \sum_{i=1}^{N} \xi_i,$$

$$\text{subj. to} \quad \begin{cases} Y_i(\sum_{j=1}^{N} k(X_i, X_j) w_j + b) \geq 1 - \xi_i, \\ \xi_i \geq 0. \end{cases}$$

The QP for SVM can be solved in the primal form, but the dual form is easier.

Dual Problem of SVM

SVM Dual problem:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j Y_i Y_j K_{ij} \quad \text{subj. to} \begin{cases} 0 \leq \alpha_i \leq C, \\ \sum_{i=1}^{N} \alpha_i Y_i = 0 \end{cases}$$

where $K_{ij} = k(X_i, X_i)$.

Solve it by a QP solver.

Note: the constraints are simpler than the primal problem.

Derivation [Exercise].

Hint: Compute the Lagrange dual function $g(\alpha, \beta)$ *from*

$$L(w, b, \xi, \alpha, \beta) = \tfrac{1}{2} \textstyle\sum_{i,j=1}^{N} w_i w_j k(X_i, X_j) + C \sum_{i=1}^{N} \xi_i$$
$$+ \textstyle\sum_{i=1}^{N} \alpha_i \{1 - Y_i (\sum_{j=1}^{N} w_j k(X_i, X_j) + b) - \xi_i\} + \sum_{i=1}^{N} \beta_i (-\xi_i).$$

KKT Conditions of SVM

KKT conditions

(1) $1 - Y_i f^*(X_i) - \xi_i^* \leq 0 \quad (\forall i)$, [primal constraints]

(2) $-\xi_i^* \leq 0 \quad (\forall i)$, [primal constraints]

(3) $\alpha_i^* \geq 0, \quad (\forall i)$, [dual constraints]

(4) $\beta_i^* \geq 0, \quad (\forall i)$, [dual constraints]

(5) $\alpha_i^*(1 - Y_i f^*(X_i) - \xi_i^*) = 0 \quad (\forall i)$, [complementary slackness]

(6) $\beta_i^* \xi_i^* = 0 \quad (\forall i)$, [complementary slackness]

(7) $\nabla_w : \quad \sum_{j=1}^n K_{ij} w_j^* - \sum_{j=1}^n \alpha_j^* Y_j K_{ij},$
$\nabla_b : \quad \sum_{j=1}^n \alpha_j^* Y_j = 0,$
$\nabla_\xi : \quad C - \alpha_i^* - \beta_i^* = 0 \quad (\forall i).$

Solution of SVM

SVM solution in dual form

$$f(x) = \sum_{i=1}^{N} \alpha_i^* Y_i k(x, X_i) + b^*.$$

(Use KKT condition (7)).

How to solve b? $\longrightarrow$ shown later.

Support Vectors I

- Complementary slackness

$$\alpha_i^*(1 - Y_i f^*(X_i) - \xi_i^*) = 0 \qquad (\forall i),$$

$$(C - \alpha_i^*)\xi_i^* = 0 \qquad (\forall i).$$

- If $\alpha_i^* = 0$, then $\xi_i^* = 0$, and

$$Y_i f^*(X_i) \geq 1. \qquad \text{[well separated]}$$

- Support vectors
  - If $0 < \alpha_i^* < C$, then $\xi_i^* = 0$ and

$$Y_i f^*(X_i) = 1. \qquad \text{[on the margin border]}$$

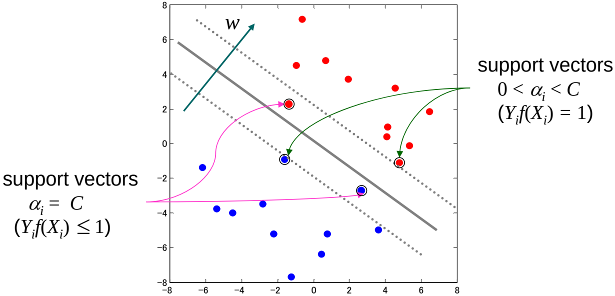
  - If $\alpha_i^* = C$,

$$Y_i f^*(X_i) \leq 1. \qquad \text{[within the margin]}$$

Support Vectors II

Sparse representation: the optimum classifier is expressed only with the support vectors.

$$f(x) = \sum_{i:\text{support vector}} \alpha_i^* Y_i k(x, X_i) + b^*$$



How to Solve b

- The optimum value of b is given by the complementary slackness.
- For any i with $0 < \alpha_i^* < C$,

$$Y_i\big(\textstyle\sum_j k(X_i, X_j) Y_j \alpha_j^* + b\big) = 1.$$

- Use the above relation for any of such i, or take the average over all of such i.

Computational Problem in Solving SVM

- The dual QP problem of SVM has N variables, where N is the sample size.
- If N is very large, say $N = 100000$, the optimization is very hard.
- Some approaches have been proposed for optimizing subsets of the variables sequentially.
  - Chunking [Vap82]
  - Osuna's method [OFG]
  - Sequential minimal optimization (SMO) [Pla99]
  - SVM$^{\text{light}}$ (`http://svmlight.joachims.org/`)

Sequential Minimal Optimization (SMO) I

- Solve small QP problems sequentially for a pair of variables (α_i, α_j).
- How to choose the pair? – Intuition from the KKT conditions is used.
    - After removing w, ξ, and β, the KKT conditions of SVM are equivalent to (see Appendix)

$$\sum_{i=1}^{N} Y_i \alpha_i^* = 0 \quad \text{and} \quad (*) \begin{cases} \alpha_i^* = 0 & \text{and} \quad Y_i f^*(X_i) \geq 1, \\ 0 < \alpha_i^* < C & \text{and} \quad Y_i f^*(X_i) = 1, \\ \alpha_i^* = C & \text{and} \quad Y_i f^*(X_i) \leq 1. \end{cases}$$

    - The conditions (*) can be checked for each data point.
    - Choose such (i, j) that at least one of them breaks (*).

Sequential Minimal Optimization (SMO) II

The QP problem for (α_i, α_j) is analytically solvable!

- For simplicity, assume $(i, j) = (1, 2)$.
- Constraint of α_1 and α_2:

$$\alpha_1 + s_{12}\alpha_2 = \gamma, \qquad 0 \le \alpha_1, \alpha_2 \le C,$$

  where $s_{12} = Y_1 Y_2$ and $\gamma = \pm \sum_{\ell \ge 3} Y_\ell \alpha_\ell$ is constat.
- Objective function:

$$\alpha_1 + \alpha_2 - \frac{1}{2}\alpha_1^2 K_{11} - \frac{1}{2}\alpha_2^2 K_{22} - s_{12}\alpha_1\alpha_2 K_{12} \\ - Y_1\alpha_1 \textstyle\sum_{j\ge 3} Y_j \alpha_j K_{1j} - Y_2 \alpha_2 \textstyle\sum_{j \ge 3} Y_j \alpha_j K_{2j} + const.$$

- This optimization is a quadratic optimization of one variable on an interval. Directly solved.

Other Approaches to Optimization of SVM

Recent studies (not a compete list).

- Solution in primal.
  - O. Chapelle [Cha07], T. Joachims, SVM$^{\text{perf}}$ [Joa06], S. Shalev-Shwartz et al. [SSSS07], etc.
- Online SVM.
  - Tax and Laskov [TL03]
  - LaSVM [BEWB05]

    `http://leon.bottou.org/projects/lasvm/`
- Parallel computation
  - Cascade SVM [GCB$^+$05]
  - Zanni et al [ZSZ06]
- Geometric approach
  - Mafrovorakis and Theodoridis [MT06].

Overview of Multiclass Classification I

- Multiclass classification:
  $(X_1, Y_1), \dots, (X_N, Y_N)$: data
  - X_i: explanatory variable
  - $Y_i \in \{C_1, \dots, C_L\}$: labels for L classes.
    e.g. Digit classification $\to L = 10$.

  Make a classifier: $h : \mathcal{X} \to \{1, 2, \dots, L\}$.

- The original SVM is applicable only to binary classification problems.

- There are some approaches to extending SVM to multiclass classification.
  - Direct construction of a large margin multiclass classifier.
  - Combination of binary classifiers.

Overview of Multiclass Classification II

Various methods (incomplete list).

- Direct approach:
  - Multiclass SVM ([CS01],[WW98], [BB99], [LLW] etc.)
  - Kernel logistic regression ([ZH02], K.Tanabe, [KDSP05])
  - and others
- Combination approach:
  - How to divide the problem
    - one-vs-rest (one-vs-all)
    - one-vs-one
    - Error correcting output code (ECOC) [DB95]
  - How to combine the binary classifiers
    - Hamming decoding
    - Bradly-Terry model ([HT98], [HWL06])

Multiclass SVM I

Multiclass SVM (Crammer & Singer 2001)

- Large margin criterion is generalized to multiclass cases.
- Efficient optimization.
- Implemented in SVM^{light}.

- Linear classifier for L-class classification
  - Data: $(X_1, Y_i), \ldots, (X_N, Y_N), \quad X_i \in \mathbb{R}^m, Y_i \in \{1, \ldots, L\}$.
  - Classifier:

$$h(x) = \arg \max_{\ell=1,\ldots,L} w_\ell^T x.$$

    L linear classifiers are used.
    (The bias term b_ℓ is omitted for simplicity.)
  - $w_\ell^T x$ ($\ell = 1, \ldots, L$) is the similarity score for the class ℓ. The class of the largest similarity is the answer of the classifier.

Multiclass SVM II

- Margin for multiclass problem:

$$\text{Margin}_i = w_{Y_i}^T X_i - \max_{\ell \neq Y_i} w_\ell^T X_i.$$

  - $W = (w_1, \ldots, w_L)$ correctly classifies the data (X_i, Y_i), if and only if $\text{Margin}_i \geq 0$.
  - The scale of the margin must be fixed.

- Primal problem of multiclass SVM:

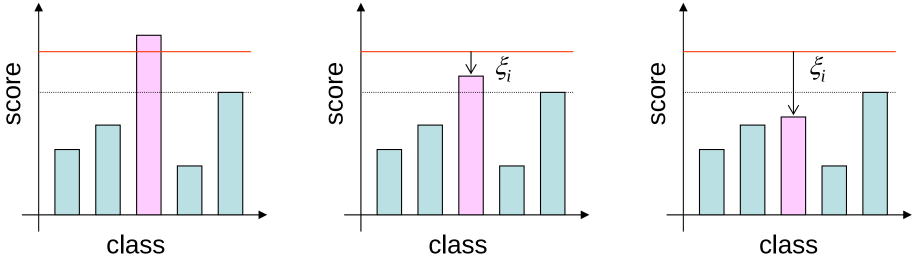
$$\min_{W,\xi} \frac{\beta}{2} \|W\|^2 + \sum_{i=1}^N \xi_i \quad \text{subj. to} \quad w_{Y_i}^T X_i + \delta_{\ell Y_i} - w_\ell^T X_i \geq 1 - \xi_i \quad (\forall \ell, i)$$

  Note: ξ_i represents the break of separability.

- # dual variable = NL. Computational cost must be reduced by some methods.

Multiclass SVM III

Meaning of margin



Combination of Binary Classifiers

- Base classifiers: make use of strong binary classifiers. *e.g.* SVM, AdaBoost, etc.
- Decomposition of a multiclass classification into binary classifications
  - 1-vs-rest

    i-class vs the other classes : L problems
  - 1-vs-1

    i-class vs j-class ($\forall i, j$) : $L(L-1)/2$ problems
  - More general approach = Error correcting output code (ECOC).

    ECOC attributes a code for each class.

class	f_1	f_2	f_3	f_4	f_5	f_6
C_1	-1	-1	-1	1	1	1
C_2	-1	1	1	-1	-1	1
C_3	1	-1	1	-1	1	-1
C_4	1	1	-1	-1	1	1

class	f_1	f_2	f_3	f_4
C_1	1	-1	-1	-1
C_2	-1	1	-1	-1
C_3	-1	-1	1	-1
C_4	-1	-1	-1	1

1-vs-rest

class	f_1	f_2	f_3	f_4	f_5	f_6
C_1	1	1	1	0	0	0
C_2	-1	0	0	1	1	0
C_3	0	-1	0	-1	0	1
C_4	0	0	-1	0	-1	-1

1-vs-1

Combining Base Classifiers

- Hamming decoding for ECOC:
  Let $W_{\ell a}$ be the code of ECOC for the class ℓ and classifier f_a $(1 \leq \ell \leq L, 1 \leq a \leq M)$.

  $$h(x) = \arg\min_{\ell} \|w_\ell - f(x)\|_{Hamming},$$

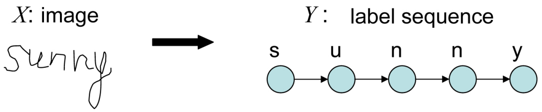
  where $f(x) = (f_1(x), \ldots, f_M(x)) \in \{\pm 1\}^M$.
  This is equivalent to

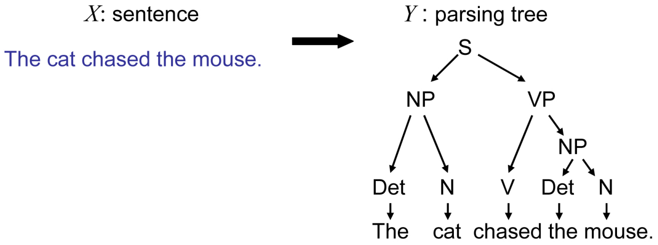
  $$h(x) = \arg\max_{\ell} \textstyle\sum_{a=1}^{M} W_{\ell a} f_a(x).$$

- In the case of one-vs-one, Hamming decoding coincides with majority vote.
- Bradly-Terry model:
  A probabilistic model for paired comparison. It can be applied when the output of $f_i(x)$ is continuous.

Structured Output

- The output of prediction may be structured object, such as label sequences (strings), trees, and graphs.





Large Margin Approach to Structured Output I

References

- Application to natural language processing [Col02].
- Max-Margin Markov Network (M^{3}N) [TGK04].
- Hidden Markov support vector machine [ATH03].

Approach

- $(X_1, Y_1), \ldots, (X_N, Y_N)$: data
  - X_i: input variable,
  - $Y_i \in \mathcal{Y}$: structured object.
- Feature vector

$$F(x, y) = (f_1(x, y), \ldots, f_M(x, y))$$

Make a classifier: $h : \mathcal{X} \to \mathcal{Y}$

$$h(x) = \arg\max_{y \in \mathcal{Y}} w^T F(x, y).$$

Large Margin Approach to Structured Output II

Formulate the problem as a multiclass classification.
Each $y \in \mathcal{Y}$ is regarded as a *class*.

- Multiclass SVM gives

$$\min_{W,\xi} \frac{\beta}{2}\|w\|^2 + \textstyle\sum_{i=1}^{N}\xi_i$$

subj. to $\quad w^T F(X_i, Y_i) + \delta_{yY_i} - w^T F(X_i, y) \geq 1 - \xi_i \quad (\forall i, y \in \mathcal{Y}).$

- Problem:
  # constrains (= # dual variables) = $|\mathcal{Y}|$. Prohibitive in many cases!
  E.g. for label sequence, $|\mathcal{Y}| = |\text{Alphabet}|^{\text{length}}$.

- The computational cost must be reduced by some methods (*e.g.* [TGK04, ATH03]).

Other Topics

- Support vector regression. [MM00]
- ν-SVM: Another formulation of soft margin. [SSWB00]
  - ν = an upper bound on the fraction of margin errors.
  - ν = the lower bound on the fraction of support vectors.
- One-class SVM: (similar to estimating a level set of density function.)
- Large margin approach to ranking.

References I

[ATH03] Y. Altun, I. Tsochantaridis, and T. Hofmann.
Hidden markov support vector machines.
In *Proceedings of the 20th International Conference on Machine Learning*, 2003.

[BB99] Erin J. Bredensteiner and Kristin P. Bennett.
Multicategory classification by support vector machines.
Computational Optimizations and Applications, 12, 1999.

[BEWB05] Antoine Bordes, Seyda Ertekin, Jason Weston, and Léon Bottou.
Fast kernel classifiers with online and active learning.
Journal of Machine Learning Research, 6:1579–1619, 2005.

[BV04] Stephen Boyd and Lieven Vandenberghe.
Convex Optimization.
Cambridge University Press, 2004.
http://www.stanford.edu/ boyd/cvxbook/.

References II

[Cha07] Olivier Chapelle.
Training a support vector machine in the primal.
Neural Computation, 19:1155–1178, 2007.

[Col02] Michael Collins.
Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms.
In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2002.

[CS01] Koby Crammer and Yoram Singer.
On the algorithmic implementation of multiclass kernel-based vector machines.
Journal of Machine Learning Research, 2:265–292, 2001.

[DB95] Thomas G. Dietterich and Ghulum Bakiri.
Solving multiclass learning problems via error-correcting output codes.
Journal of Artificial Intelligence Research, 2:263–286, 1995.

References III

[GCB+05] Hans Peter Graf, Eric Cosatto, Léon Bottou, Igor Dourdanovic, and Vladimir Vapnik.
Parallel support vector machines: The Cascade SVM.
In Lawrence Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2005.

[HT98] T. Hastie and R. Tibshirani.
Classification by pairwise coupling.
The Annals of Statistics, 26(1):451–471, 1998.

[HWL06] Tzu-Kuo Huang, Ruby C. Weng, and Chih-Jen Lin.
Generalized Bradly-Terry models and multi-class probability estimates.
Journal of Machine Learning Research, 7:85–115, 2006.

References IV

[Joa06] Thorsten Joachims.
Training linear svms in linear time.
In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.

[KDSP05] S. S. Keerthi, K. B. Duan, S. K. Shevade, and A. N. Poo.
A fast dual algorithm for kernel logistic regression.
Machine Learning, 61(1–3):151–165, 2005.

[LLW] Y. Lee, Y. Lin, and G. Wahba.
Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data.
Journal of the American Statistical Association, 99.

[MM00] O. L. Mangasarian and D. R. Musicant.
Robust linear and support vector regression.
IEEE Trans. Pattern Analysis Machine Intelligence, 22, 2000.

References V

[MT06] M.E. Mavroforakis and S. Theodoridis.
A geometric approach to support vector machine (SVM) classification.
IEEE Trans. Neural Networks, 17(3), 2006.

[OFG] Edgar Osuna, Robert Freund, and Federico Girosi.
An improved training algorithm for support vector machines.
In *Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing (IEEE NNSP 1997)*, pages 276–285.

[Pla99] John Platt.
Fast training of support vector machines using sequential minimal optimization.
In Bernhard Schölkopf, Cristopher Burges, and Alexander Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 185–208. MIT Press, 1999.

References VI

[Shi08] Yuichi Shiraishi.
Game-theoretical and statistical study on combination of binary classifiers for multi-class classification.
Ph.D. thesis, Department of Statistical Science, The Graduate University for Advanced Studies, 2008.

[SSSS07] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro.
Pegasos: Primal estimated sub-gradient solver for svm.
In *Proc. International Conerence of Machine Learning*, 2007.

[SSWB00] B. Schölkopf, A. Smola, R. C. Williamson, and P. L. Bartlett.
New support vector algorithms.
Neural Computation, 12:1207–1245, 2000.

[TGK04] Ben Taskar, Carlos Guestrin, and Daphne Koller.
Max-margin markov networks.
In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

References VII

[TL03] D.M.J. Tax and P. Laskov.
Online svm learning: from classification to data description and back.
In *Proceddings of IEEE 13th Workshop on Neural Networks for Signal Processing (NNSP2003)*, pages 499–508, 2003.

[Vap82] Vladimir N. Vapnik.
Estimation of Dependences Based on Empirical Data.
Springer-Verlag, 1982.

[WW98] J. Weston and C. Watkins.
Multi-class support vector machines.
Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, 1998.

[ZH02] Ji Zhu and Trevor Hastie.
Kernel logistic regression and the import vector machine.
14:1081–1088, 2002.

References VIII

[ZSZ06] Luca Zanni, Thomas Serafini, and Gaetano Zanghirati.
Parallel software for training large scale support vector machines on multiprocessor systems.
Journal of Machine Learning Research, 7:1467–1492, 2006.

Appendix: Proof of KKT condition

Proof.

- x^* is primal-feasible by the first two conditions.
- From $\lambda_i^* \geq 0$, $L(x, \lambda^*, \nu^*)$ is convex (and differentiable).
- The last condition $\nabla_x L(x^*, \lambda^*, \nu^*) = 0$ implies x^* is a minimizer.
- It follows

$$
\begin{aligned}
g(\lambda^*, \nu^*) &= \inf_{x \in \mathcal{D}} L(x, \lambda^*, \nu^*) \qquad \text{[by definition]} \\
&= L(x^*, \lambda^*, \nu^*) \qquad [x^*\text{: minimizer}] \\
&= f(x^*) + \textstyle\sum_{i=1}^{\ell} \lambda_i^* h_i(x^*) + \sum_{j=1}^{m} \nu_j^* r_j(x^*) \\
&= f(x^*) \qquad \text{[complementary slackness and } r_j(x^*) = 0\text{].}
\end{aligned}
$$

- Strong duality holds, and x^* and (λ^*, ν^*) must be the optimizers.

Appendix: KKT conditions revisited I

- β and w can be removed by

$$\begin{aligned} \nabla_\xi : \quad & \beta_i^* = C - \alpha_i^* \quad (\forall i), \\ \nabla_w : \quad & \textstyle\sum_{j=1}^n K_{ij} w_j^* = \sum_{j=1}^n \alpha_j^* Y_j K_{ij} \quad (\forall i). \end{aligned}$$

- From KKT (4) and (6),

$$\alpha_i^* \leq C, \qquad \xi_i^*(C - \alpha_i^*) = 0 \quad (\forall i).$$

- The KKT conditions are equivalent to

(a) $1 - Y_i f^*(X_i) - \xi_i^* \leq 0 \quad (\forall i)$,
(b) $\xi_i^* \geq 0 \quad (\forall i)$,
(c) $0 \leq \alpha_i^* \leq C \quad (\forall i)$,
(d) $\alpha_i^*(1 - Y_i f^*(X_i) - \xi_i^*) = 0 \quad (\forall i)$,
(e) $\xi_i^*(C - \alpha_i^*) = 0 \quad (\forall i)$,
(f) $\sum_{i=1}^N Y_i \alpha_i^* = 0$.

and $\beta_i = C - \alpha_i^*$, $\sum_{j=1}^n K_{ij} w_j^* = \sum_{j=1}^n \alpha_j^* Y_j K_{ij}$.

Appendix: KKT conditions revisited II

- We can further remove ξ.
  - Case $\alpha_i^* = 0$:

    From (e), $\xi_i^* = 0$. Then, from (a), $Y_i f^*(X_i) \geq 1$.
  - Case $0 < \alpha_i^* < C$:

    From (e), $\xi_i^* = 0$. From (d), $Y_i f^*(X_i) = 1$.
  - Case $\alpha_i^* = C$:

    From (d) and (b), $\xi_i^* = 1 - Y_i f^*(X_i) \geq 0$.

  Note in all cases, (a) and (b) are satisfied.

- The KKT conditions are equivalent to

$\sum_{i=1}^N Y_i \alpha_i^* = 0,$ and

$$\begin{cases} \alpha_i^* = 0 & \Rightarrow \quad Y_i f^*(X_i) \geq 1, \quad (\xi_i^* = 0) \\ 0 < \alpha_i^* < C & \Rightarrow \quad Y_i f^*(X_i) = 1, \quad (\xi_i^* = 0) \\ \alpha_i^* = C & \Rightarrow \quad Y_i f^*(X_i) \leq 1, \quad (\xi_i^* = 1 - Y_i f^*(X_i)). \end{cases}$$