统计机器学习

（小班研讨）

主讲教师： 刘峤

第4章　支持向量机与核方法

Support Vector Machines and Kernel Methods

Soft-Margin Support Vector Machine

Estimate the Margin

- Min-max problem

$$\operatorname*{argmax}_{\mathbf{w},b} \ \min_{\mathbf{x}_i \in D} \frac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\sqrt{\sum_{i=1}^{d} \mathbf{w}_i^2}}$$

$$\text{subject to } \ \forall \mathbf{x}_i \in D : y_i \left(\mathbf{x}_i \cdot \mathbf{w} + b \right) \geq 0$$

↓ **This is going to be a problem!**

- 线性可分支持向量机的最优化问题 **What should we do?**

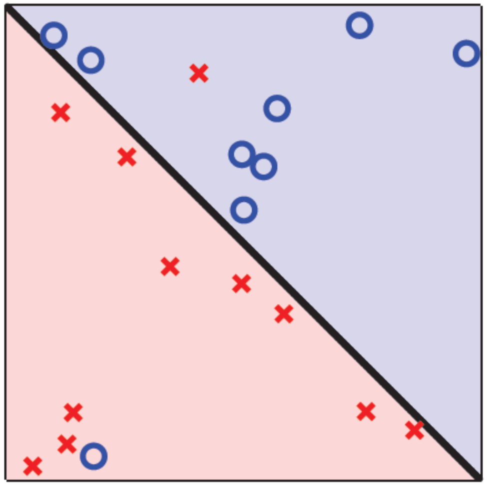
$$\operatorname*{argmin}_{\mathbf{w},b} \sum_{i=1}^{d} \mathbf{w}_i^2$$

$$\text{subject to } \ \forall \mathbf{x}_i \in D : y_i \left(\mathbf{x}_i \cdot \mathbf{w} + b \right) \geq 1$$

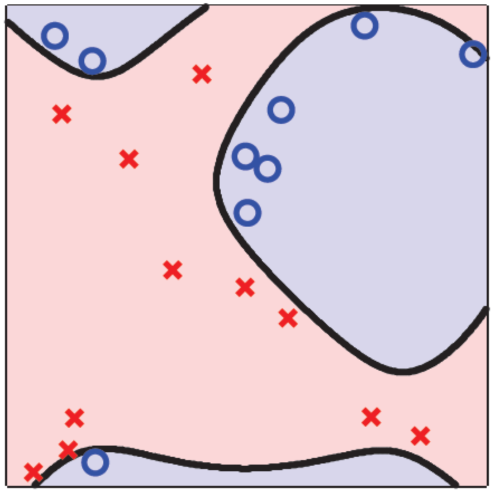
- How to solve it? **Quadratic Programming?**

Cons of Hard-Margin SVM

- **Recall: SVM can still overfit**



$\mathbf{\Phi}_1$



$\mathbf{\Phi}_4$

- part of reasons: Φ could be overly powerful
- other part: **separable (shatter)**
  - have power to **overfit to noise**

Give Up on Some Examples

- **Basic idea：** **give up** on some noisy examples

- Idea 1: Minimize $\mathbf{w}^T\mathbf{w}$
  – while minimizing number of training set errors.
- **problem**: two things to minimize …
  – makes for an ill-defined optimization

Give Up on Some Examples

- **Basic idea : give up** on some noisy examples

- Idea 1.1: Minimize $\mathbf{w}^T\mathbf{w} + C(\#\text{train errors})$

  non-linear

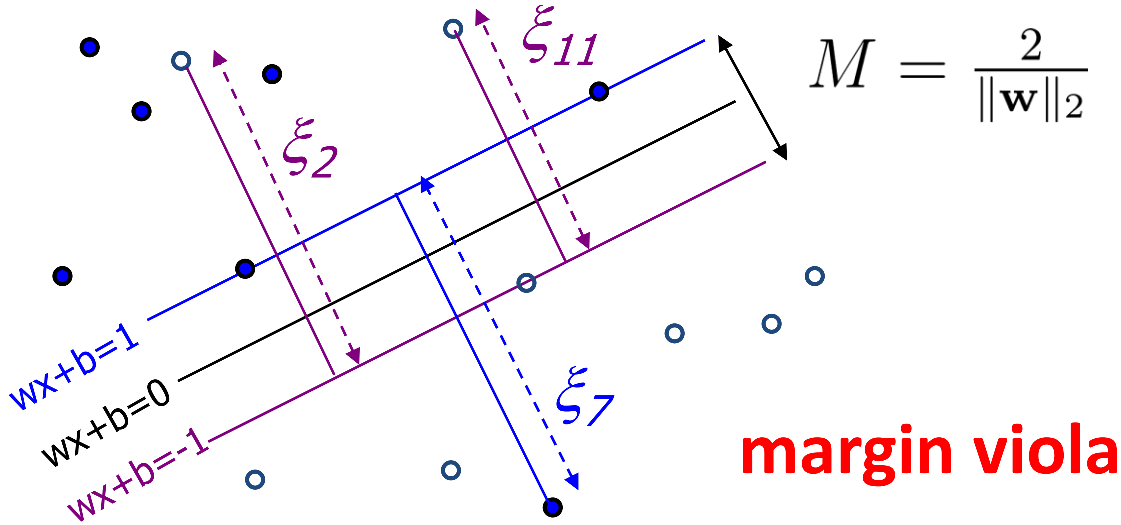
  – C is called the **Tradeoff** parameter

  – There's a serious practical problem that's about to make us reject this approach. Can you guess what it is?

    - ⁎ Can't be expressed as a QP problem.
    - ⁎ Doesn't distinguish between :
      - ⁎ disastrous errors and near misses

So… any other ideas?

Soft-Margin SVM



margin violation

- Idea 2.0: Minimize

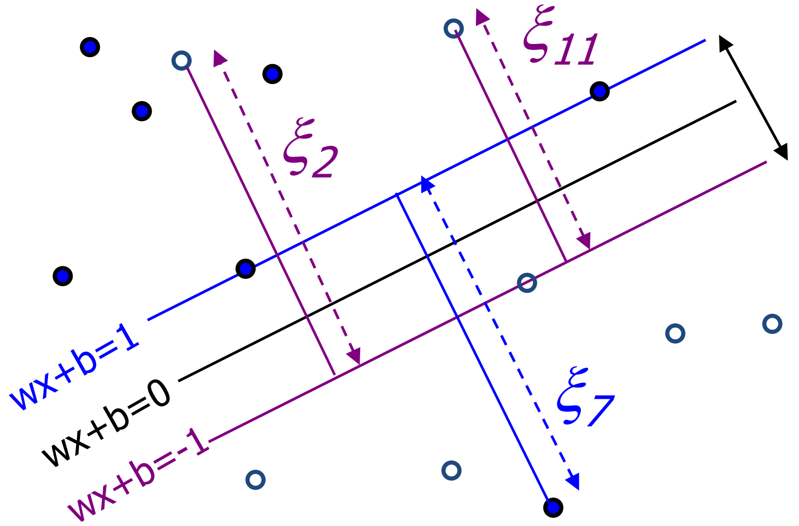
  w.w + C (distance of error points to their correct place)

- What should our quadratic optimization criterion be?

$$\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i$$

Soft-Margin SVM

$$\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i$$



$$M = \frac{2}{\|\mathbf{w}\|_2}$$

- How many constraints will we have? ***N*****data**
- What should they be?

$$\xi_i \geq 0 \text{ for all n}$$

$$\left.\begin{array}{l} \mathbf{w}^T\mathbf{x}_i + b \geq +1 - \xi_i \;\text{ if } y_i = +1 \\ \mathbf{w}^T\mathbf{x}_i + b \leq -1 + \xi_i \;\text{ if } y_i = -1 \end{array}\right\} \; y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$$

linear constraints

Soft-Margin SVM

- **Our quadratic optimization criterion**

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

$$s.t. \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$$

There's a bug in this QP.

Can you spot it?

- **What should the constraints be?**

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

$$s.t. \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$$

How do we solve it?

$$\text{and} \quad \xi_i \geq 0 \text{ for all i}$$

Soft-Margin SVM

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

$$s.t.\ \ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \ \ \text{and} \ \ \xi_i \geq 0 \text{ for all i}$$

- **parameter C: trade-off of large margin & margin violation**
  - **large C: want less margin violation**
  - **small C: want large margin**
- **How many constraints will we have?**
  - **k+1+N variables, 2N constraints**
- **next: remove dependence on k by primal → dual**

Lagrange Dual

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

$$s.t.\;\; y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \;\text{ and }\; \xi_i \geq 0 \text{ for all n}$$

- **Lagrange function with Lagrange multipliers** α_i and β_i

$$\mathcal{L}(b,\mathbf{w},\xi,\alpha,\beta) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i(1 - \xi_i - y_i(\mathbf{w}^T\mathbf{x}_i + b)) + \sum_{i=1}^{N}\beta_i(-\xi_i)$$

- **want: Lagrange dual**

$$\max_{\alpha_i \geq 0, \beta_i \geq 0} \left(\min_{b,\mathbf{w},\xi} \mathcal{L}(b,\mathbf{w},\xi,\alpha,\beta) \right)$$

Simplify ξ_i and β_i

$$\mathcal{L}(b, \mathbf{w}, \xi, \alpha, \beta) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i(1 - \xi_i - y_i(\mathbf{w}^T\mathbf{x}_i + b)) + \sum_{i=1}^{N}\beta_i(-\xi_i)$$

$$\frac{\partial\mathcal{L}}{\partial\xi_i} = C - \alpha_i - \beta_i = 0 \Rightarrow \beta_i = C - \alpha_i$$ **implicit constraint**

$$\alpha_i \geq 0, \beta_i \geq 0 \Rightarrow 0 \leq \alpha_i \leq C$$ **explicit constraint**

- **no loss of optimality if solving with implicit & explicit constraint**

$$\max_{0\leq\alpha_i\leq C,\beta_i=C-\alpha_i}\left(\min_{b,\mathbf{w}} \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{N}\alpha_i(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))\right)$$

Note: ξ_i can also be removed in this way :-)

Lagrange Dual

$$\max_{0\leq\alpha_i\leq C, \beta_i = C-\alpha_i} \left(\min_{b,\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{N} \alpha_i(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)) \right)$$

- **Looks familiar?**
  - – inner problem same as hard-margin SVM
  - – no loss of optimality if solving with constraints :

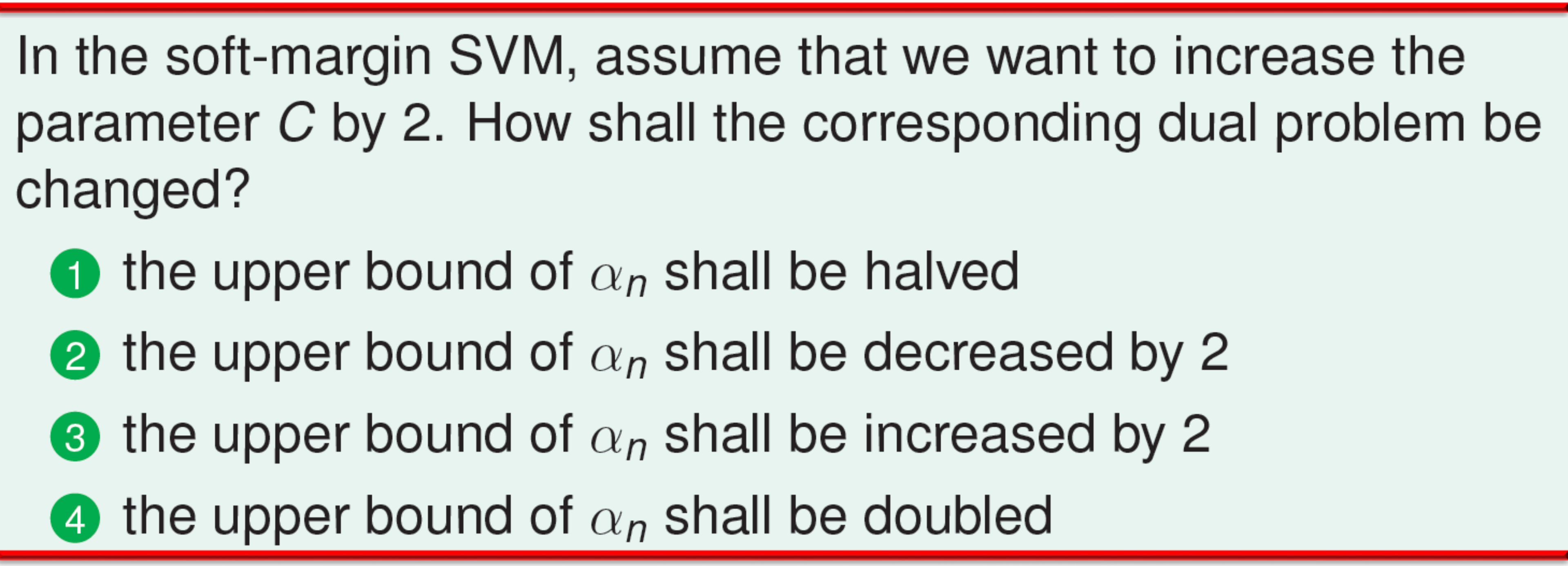
$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}, b, \alpha) = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i = 0 \quad \Rightarrow \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

$$\nabla_{b}\mathcal{L}(\mathbf{w}, b, \alpha) = -\sum_{i=1}^{N} \alpha_i y_i = 0 \qquad \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0$$

Standard Soft-Margin SVM Dual

- 对偶问题： $$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i \cdot \mathbf{x}_j) - \sum_{i=1}^{N}\alpha_i$$
  $$\text{s.t.} \quad \sum_{i=1}^{N}\alpha_i y_i = 0; \quad 0 \le \alpha_i \le C,\ i = 1, 2, \cdots, N$$
- 隐含约束条件： $$\mathbf{w} = \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i$$
  $$\beta_i = C - \alpha_i,\ i = 1, 2, \cdots, N$$
- 与hard-margin方法相比，区别在哪里?
  - 为 α_i 引入了一个上界
  - 约束条件变为：N variables & 2N + 1 constraints

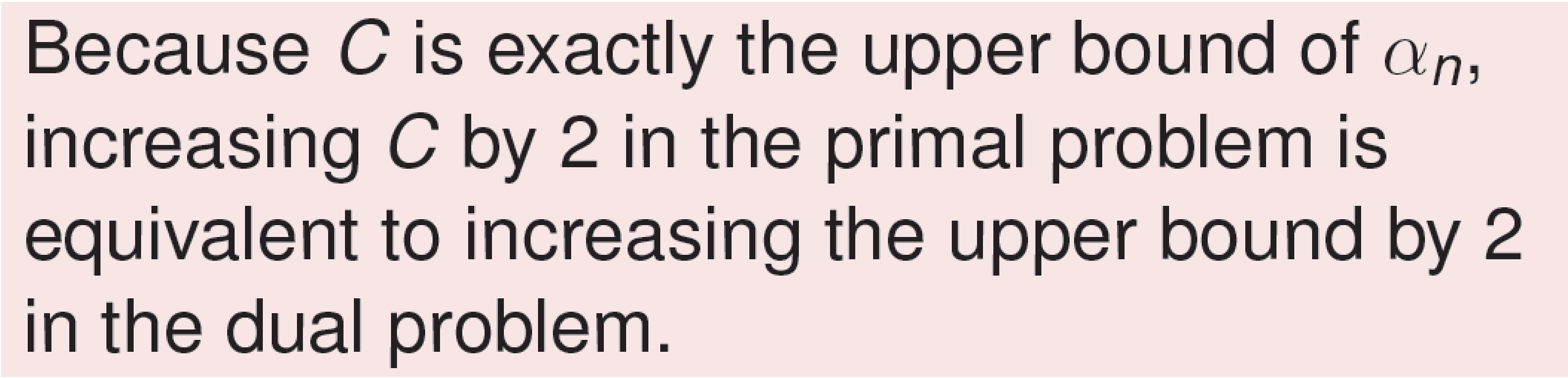
Quize



In the soft-margin SVM, assume that we want to increase the parameter C by 2. How shall the corresponding dual problem be changed?

1. the upper bound of α_n shall be halved
2. the upper bound of α_n shall be decreased by 2
3. the upper bound of α_n shall be increased by 2
4. the upper bound of α_n shall be doubled

Reference Answer: 3



Because C is exactly the upper bound of α_n, increasing C by 2 in the primal problem is equivalent to increasing the upper bound by 2 in the dual problem.

Solving for *b*

- hard-margin SVM : complementary slackness

$$\alpha_i(1 - y_i(\mathbf{w}^T\mathbf{x} + b)) = 0$$

$$\forall \alpha_j > 0 \text{ (i.e. SV) } : \; b = y_j - \mathbf{w}^T\mathbf{x}_j$$

- soft-margin SVM : complementary slackness?

$$(1) \;\; \alpha_i(1 - \xi_i - y_i(\mathbf{w}^T\mathbf{x}_i + b) = 0$$

$$\Rightarrow \forall \alpha_j > 0 \text{ (i.e. SV) } : \; b = y_j - y_j\xi_i - \mathbf{w}^T\mathbf{x}_j$$

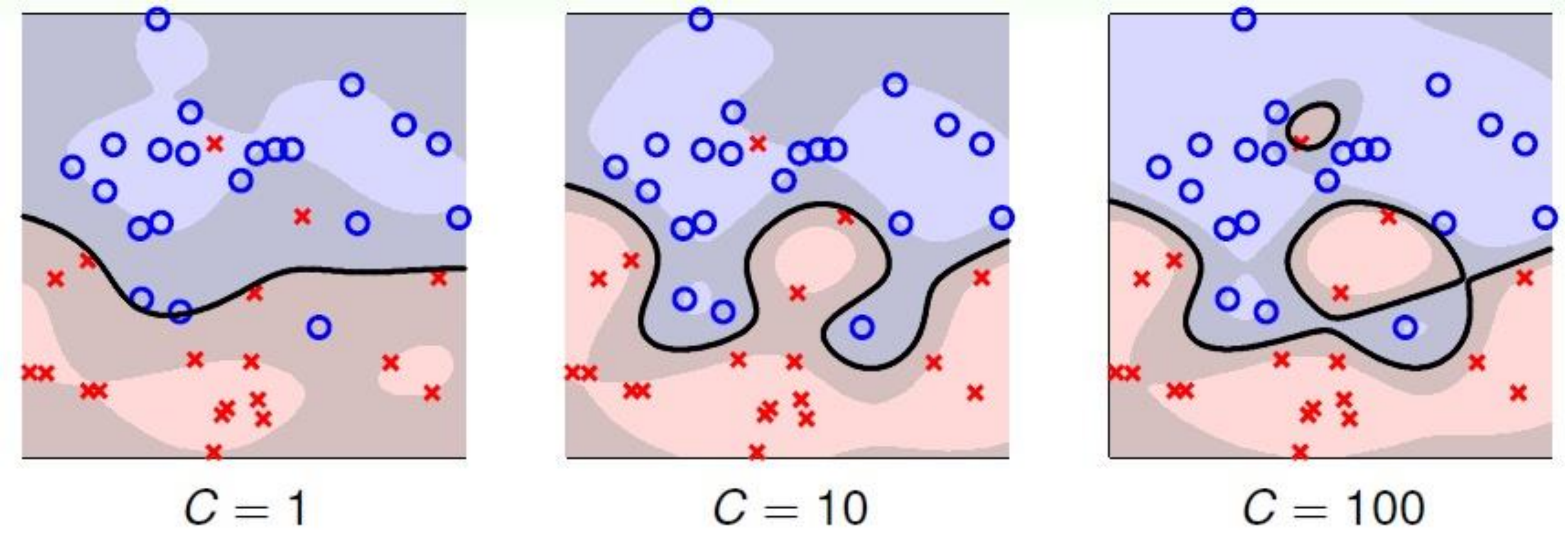
$$(2) \; (C - \alpha_i)\xi_i = 0 \;\; \Rightarrow \text{if } \alpha_i < C, \text{ then } \xi_i = 0$$ **free SV**

- solve unique b with free SV $(\mathbf{x}_j, y_j)$:

$$b^* = y_j - \sum_{i \in SV} \alpha_i y_i \mathbf{x}_i^T \mathbf{x}_j$$

Soft-Margin Gaussian SVM in Action

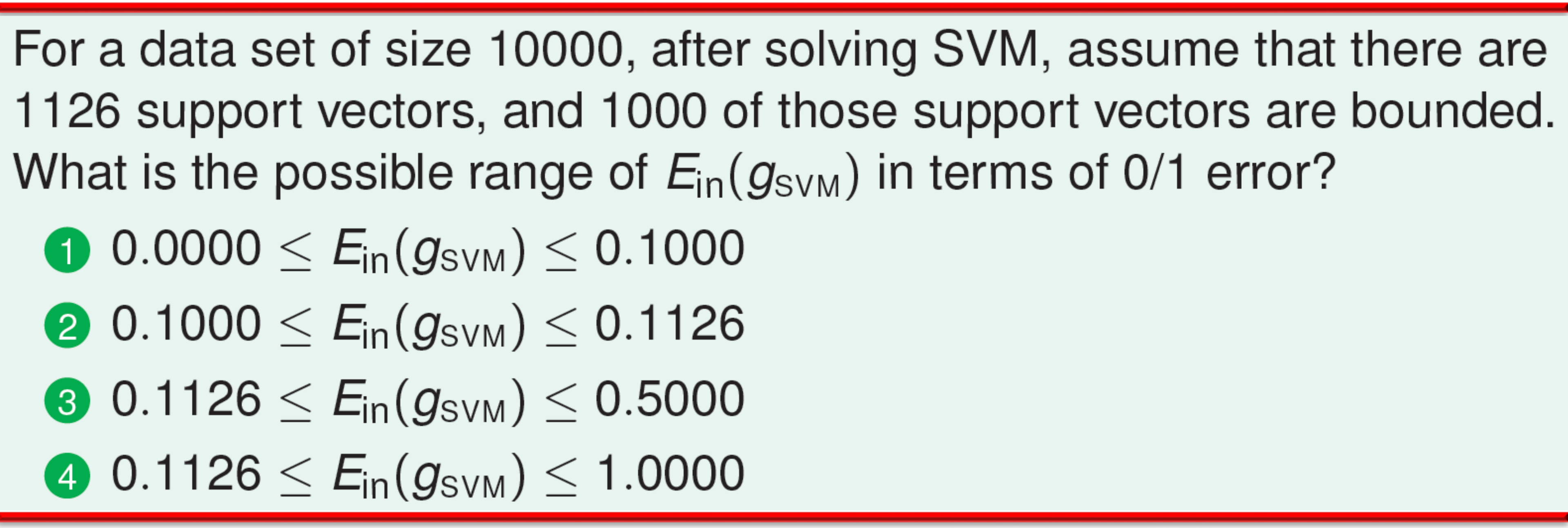
- How do we determine the appropriate value for C ?
  - Recall: trade-off of large margin & margin violation



Guassian kernel/soft-margin SVM在不同参数C下的实验结果

- large C $\Rightarrow$ less noise tolerance $\Rightarrow$?   **overfit**

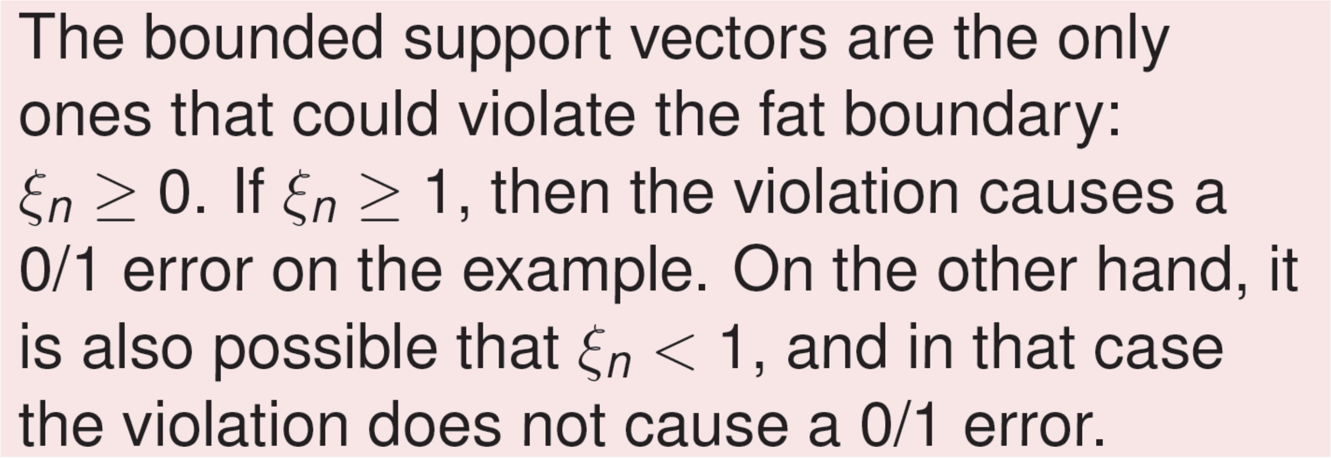
Quize



For a data set of size 10000, after solving SVM, assume that there are 1126 support vectors, and 1000 of those support vectors are bounded. What is the possible range of $E_{\text{in}}(g_{\text{SVM}})$ in terms of 0/1 error?

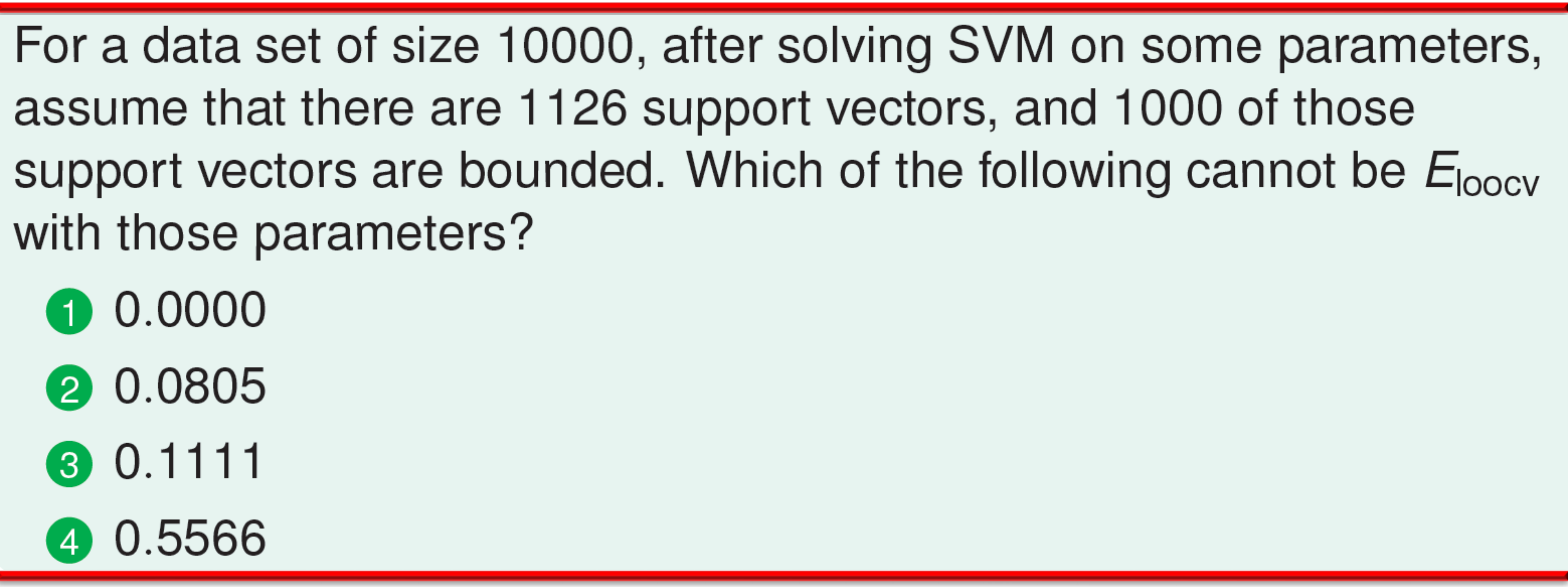
1. $0.0000 \leq E_{\text{in}}(g_{\text{SVM}}) \leq 0.1000$
2. $0.1000 \leq E_{\text{in}}(g_{\text{SVM}}) \leq 0.1126$
3. $0.1126 \leq E_{\text{in}}(g_{\text{SVM}}) \leq 0.5000$
4. $0.1126 \leq E_{\text{in}}(g_{\text{SVM}}) \leq 1.0000$

Reference Answer: 1



The bounded support vectors are the only ones that could violate the fat boundary: $\xi_n \geq 0$. If $\xi_n \geq 1$, then the violation causes a 0/1 error on the example. On the other hand, it is also possible that $\xi_n < 1$, and in that case the violation does not cause a 0/1 error.

Quize



For a data set of size 10000, after solving SVM on some parameters, assume that there are 1126 support vectors, and 1000 of those support vectors are bounded. Which of the following cannot be E_{loocv} with those parameters?

1. 0.0000
2. 0.0805
3. 0.1111
4. 0.5566

Reference Answer: 4

Note that the upper bound of E_{loocv} is 0.1126.

Summarization

Hard-Margin Primal

$$\min_{b,\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$\text{s.t.} \quad y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1$$

Soft-Margin Primal

$$\min_{b,\mathbf{w},\xi} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n$$

$$\text{s.t.} \quad y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1 - \xi_n, \xi_n \geq 0$$

Hard-Margin Dual

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\boldsymbol{\alpha}^T \mathrm{Q} \boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha}$$

$$\text{s.t.} \quad \mathbf{y}^T\boldsymbol{\alpha} = 0$$

$$0 \leq \alpha_n$$

Soft-Margin Dual

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\boldsymbol{\alpha}^T \mathrm{Q} \boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha}$$

$$\text{s.t.} \quad \mathbf{y}^T\boldsymbol{\alpha} = 0$$

$$0 \leq \alpha_n \leq C$$

- soft-margin preferred in practice !
  - linear: LIBLINEAR; non-linear: LIBSVM

Kernel Logistic Regression

Soft-Margin SVM as Regularized Model

- Soft-Margin SVM : $\min\limits_{b,\mathbf{w},\xi} \left\{ \frac{1}{2}\mathbf{w}^T\mathbf{w} + \boxed{C \sum\limits_{i=1}^{N} \xi_i} \right\}$ **penalty**

$$s.t. \;\; y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \;\; \text{and} \;\; \xi_i \geq 0 \text{ for all n}$$

- ξ_i : record margin violation
  - $(\mathbf{x}_i, y_i)$ not violating margin: $\xi_i = 0$
  - $(\mathbf{x}_i, y_i)$ violating margin: $\xi_i = 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)$
  - on any $(\mathbf{w}, b)$: $\xi_i = \max\left\{0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right\}$
- unconstrained form of soft-margin SVM **L2 regularization**

$$\min_{b,\mathbf{w},\xi} \boxed{\frac{1}{2}\mathbf{w}^T\mathbf{w}} + C \sum_{i=1}^{N} \max\left\{0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right\}$$

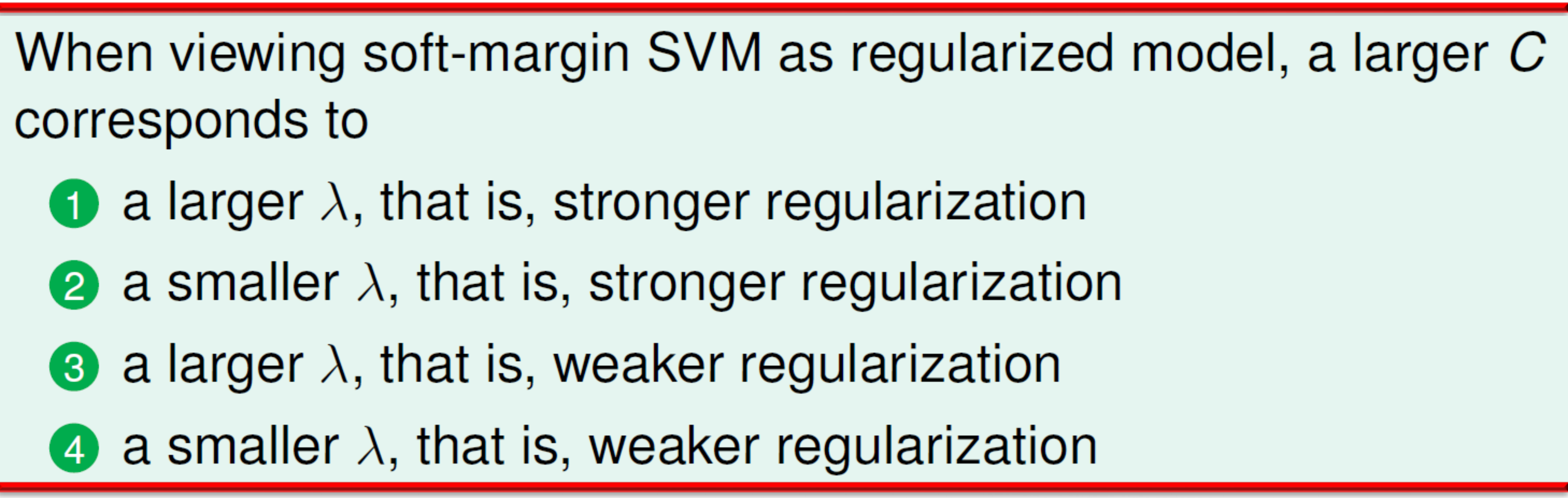
in sample error

SVM as Regularized Model

- Soft-Margin SVM : $\min_{\mathbf{w}} \; C \sum_{i=1}^{N} \mathrm{err}' + \frac{1}{2}\mathbf{w}^T\mathbf{w}$
- L2 regularization : $\min_{\mathbf{w}} \; \frac{1}{N} \sum_{i=1}^{N} \mathrm{err} + \frac{\lambda}{N}\mathbf{w}^T\mathbf{w}$

- large margin $\Longleftrightarrow$ L2 regularization of short **w**
- larger C $\Longleftrightarrow$ smaller λ $\Longleftrightarrow$ less regularization
- soft margin? $\Longleftrightarrow$ special error'
- viewing SVM as regularized model:
  - allows extending/connecting to other learning models

Quize



When viewing soft-margin SVM as regularized model, a larger C corresponds to

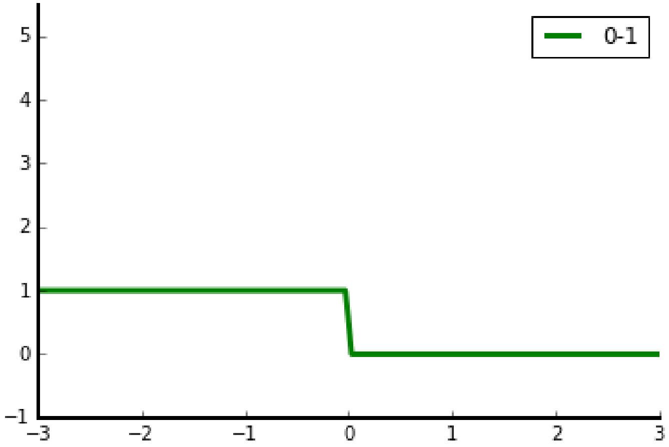
1. a larger λ, that is, stronger regularization
2. a smaller λ, that is, stronger regularization
3. a larger λ, that is, weaker regularization
4. a smaller λ, that is, weaker regularization

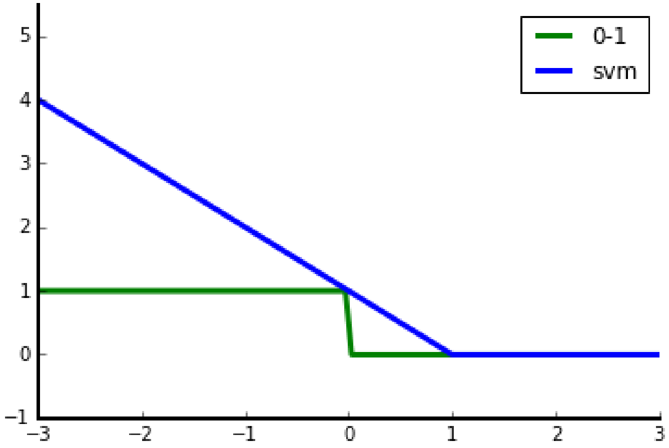
Reference Answer: 1

Note that $C \propto \frac{1}{\lambda}$, So larger C corresponds to smaller λ

Algorithmic Error Measure of SVM

$$\min_{b,\mathbf{w},\xi} \ \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N} \max\left\{0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right\}$$





- linear score $s = \mathbf{w}^T\mathbf{x}_i + b$

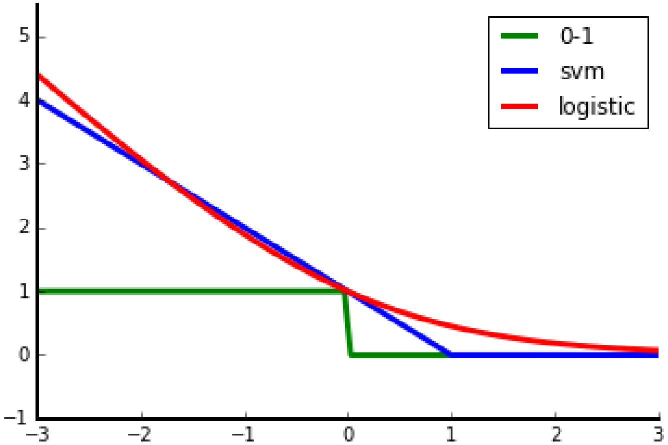
$$E_{0-1} = ind(ys < 0)$$

$$E_{svm} = \max(1 - ys, 0)$$

- **hinge** error measure
  - **upper bound** of E_{0-1}

Connection between SVM and Logistic Regression

$$\min_{b,\mathbf{w},\xi} \ \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\max\left\{0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right\}$$



- linear score $s = \mathbf{w}^T\mathbf{x}_i + b$

$$E_{svm} = \max(1 - ys, 0)$$

$$E_{logit} = \log_2(1 + exp(-ys))$$

if $ys \to +\infty \ : \ E_{svm} \to 0$ if $ys \to -\infty \ : \ E_{svm} \to -ys$

if $ys \to +\infty \ : \ E_{logit} \to 0$ if $ys \to -\infty \ : \ E_{logit} \to -ys$

- SVM $\approx$ L2-regularized logistic regression

Recap: Logistic Classification

- Logistic regression for binary classification $y \in \{-1, 1\}$

$$P(Y = 1|\mathbf{x}, \mathbf{w}) = \frac{1}{1+\exp\{-\mathbf{w}^\top \mathbf{x}\}} = \frac{1}{1+\exp\{-y\mathbf{w}^\top \mathbf{x}\}}$$

$$P(Y = -1|\mathbf{x}, \mathbf{w}) = 1 - P(Y = 1|\mathbf{x}, \mathbf{w}) = \frac{1}{1+\exp\{-y\mathbf{w}^\top \mathbf{x}\}}$$

$$log(\frac{P(Y=1|\mathbf{x},\mathbf{w})}{P(Y=-1|\mathbf{x},\mathbf{w})}) = \mathbf{w}^\top \mathbf{x}$$

- The log likelihood of the data ℓ(w) is:

$$\log(P(D_Y|D_X, \mathbf{w})) = \log\left(\prod_i \frac{1}{1+\exp\{-y_i \mathbf{w}^\top \mathbf{x}_i\}} \right)$$

$$= -\sum_i \log(1 + \exp\{-y_i \mathbf{w}^\top \mathbf{x}_i\})$$

Computing MLE

- The gradient of our objective is given by:

$$\nabla_{\mathbf{w}}\, \ell(\mathbf{w}) = \frac{\partial \log(P(D_Y|D_X,\mathbf{w}))}{\partial \mathbf{w}} \quad = \frac{\partial - \sum_i \log(1+\exp\{-y_i \mathbf{w}^\top \mathbf{x}_i\})}{\partial \mathbf{w}}$$

$$= \sum_i \frac{y_i \mathbf{x}_i \exp\{-y_i \mathbf{w}^\top \mathbf{x}_i\}}{1+\exp\{-y_i \mathbf{w}^\top \mathbf{x}_i\}} \quad = \sum_i y_i \mathbf{x}_i (1 - P(y_i|\mathbf{x}_i, \mathbf{w}))$$

- Gradient ascent update is

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \eta_t \nabla_{\mathbf{w}} \ell(\mathbf{w})$$

- Where $\eta_t > 0$ is the update rate.
- iterate until change is in parameters is smaller than some tolerance

Key behind Kernel Trick

- one key behind kernel trick: optimal $\mathbf{w}^* = \sum_{i=1}^{N} \beta_i \mathbf{z}_i$

$$\mathbf{w}^{*T}\mathbf{z} = \sum_{i=1}^{N} \beta_i \mathbf{z}_i \mathbf{z} = \sum_{i=1}^{N} \beta_i K(\mathbf{x}_i, \mathbf{x})$$

- PLA： $\mathbf{w}_{PLA} = \sum_{i=1}^{N} (\alpha_i y_i) \mathbf{x}_i$ α_i by # mistake corrections
- SVM： $\mathbf{w}_{SVM} = \sum_{i=1}^{N} (\alpha_i y_i) \mathbf{x}_i$ α_i from dual solutions
- LogR： $\mathbf{w}_{LogR} = \sum_{i=1}^{N} (\alpha_i y_i) \mathbf{x}_i$ α_i by total SGD moves
- Question： when can optimal w* be represented by z_n?

Representer Theorem

- Claim: for any L2-regularized linear model

$$\min_{\mathbf{w}} \quad \frac{\lambda}{N}\mathbf{w}^T\mathbf{w} + \frac{1}{N}\sum_{i=1}^{N} err(y_i, \mathbf{w}^T\mathbf{z}_i)$$

- optimal $\mathbf{w}^* = \sum_{i=1}^{N} \beta_i \mathbf{z}_i$

- let optimal $\mathbf{w}^* = \mathbf{w}_{\mathbf{z}} + \mathbf{w}_p$ where

$$\mathbf{w}_{\mathbf{z}} \in \mathrm{span}(\mathbf{z}_i) \text{ and } \mathbf{w}_p \perp \mathrm{span}(\mathbf{z}_i) \quad \mathbf{w}_p = 0?$$

- if $\mathbf{w}_p \neq 0$, consider : $\mathbf{w}_{\mathbf{z}}$
  - of same err as $\mathbf{w}^*$:

$$err(y_i, \mathbf{w}^{*T}\mathbf{z}_i) = err(y_i, (\mathbf{w}_{\mathbf{z}} + \mathbf{w}_p)^T\mathbf{z}_i)$$

Representer Theorem

$$\min_{\mathbf{w}} \quad \frac{\lambda}{N}\mathbf{w}^T\mathbf{w} + \frac{1}{N}\sum_{i=1}^{N} err(y_i, \mathbf{w}^T\mathbf{z}_i)$$

- let optimal $\mathbf{w}^* = \mathbf{w}_{\mathbf{z}} + \mathbf{w}_p$ where

$$\mathbf{w}_{\mathbf{z}} \in \text{span}(\mathbf{z}_i) \text{ and } \mathbf{w}_p \perp \text{span}(\mathbf{z}_i) \quad \mathbf{w}_p = 0?$$

- if $\mathbf{w}_p \neq 0$, consider : $\mathbf{w}_{\mathbf{z}}$
  - of same err as $\mathbf{w}^*$: $err(y_i, \mathbf{w}^{*T}\mathbf{z}_i) = err(y_i, (\mathbf{w}_{\mathbf{z}} + \mathbf{w}_p)^T\mathbf{z}_i)$
  - of smaller regularizer as $\mathbf{w}^*$:

$$\mathbf{w}^{*T}\mathbf{w}^* = \mathbf{w}_{\mathbf{z}}^T\mathbf{w}_{\mathbf{z}} + 2\mathbf{w}_{\mathbf{z}}^T\mathbf{w}_p + \mathbf{w}_p^T\mathbf{w}_p > \mathbf{w}_{\mathbf{z}}^T\mathbf{w}_{\mathbf{z}}$$

- $\mathbf{w}_{\mathbf{z}}$ more optimal than $\mathbf{w}^*$ **--- contradiction!**
- **any L2-regularized linear model can be kernelized!**

Kernelized Logistic Regression (KLR)

- Logistic Regression:

$$p(y|\mathbf{z}) = \frac{1}{1+\exp(-y\mathbf{w}^T\mathbf{z})}$$

- Loss function ：

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}\log\frac{1}{1+\exp(-y_i\mathbf{w}^T\mathbf{z}_i)} - \frac{\lambda}{N}\mathbf{w}^T\mathbf{w}$$

- solving L2-regularized logistic regression (Primal Problem)

$$\min_{\mathbf{w}} \ \frac{\lambda}{N}\mathbf{w}^T\mathbf{w} + \frac{1}{N}\sum_{i=1}^{N}\log\left(1 + exp(-y_i\mathbf{w}^T\mathbf{z}_i)\right)$$

- yields optimal solution $\mathbf{w}^* = \sum_{i=1}^{N}\beta_i\mathbf{z}_i$

Kernelized Logistic Regression (KLR)

- Primal Problem

$$\min_{\mathbf{w}} \ \frac{\lambda}{N}\mathbf{w}^T\mathbf{w} + \frac{1}{N}\sum_{i=1}^{N} \log\left(1 + exp(-y_i\mathbf{w}^T\mathbf{z}_i)\right)$$

- with out loss of generality, can solve for optimal β

$$\min_{\beta} \ \frac{\lambda}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\beta_i\beta_j K(\mathbf{x}_i,\mathbf{x}_j) + \frac{1}{N}\sum_{i=1}^{N}\log\left(1 + exp\left(-y_i\sum_{j=1}^{N}\beta_j K(\mathbf{x}_j,\mathbf{x}_i)\right)\right)$$

- How to solve it ?   SGD for unconstrained optimization

- kernel logistic regression:
  - use **representer theorem** for kernel trick
  - on **L2-regularized** logistic regression

KLR : Another View

$$\min_{\beta} \frac{\lambda}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{N} \sum_{i=1}^{N} \log \left(1 + exp \left(-y_i \sum_{j=1}^{N} \beta_j K(\mathbf{x}_j, \mathbf{x}_i) \right) \right)$$

$\sum_{j=1}^{N} \beta_j K(\mathbf{x}_j, \mathbf{x}_i)$: inner product between variables β and transformed data : $(K(\mathbf{x}_1, \mathbf{x}_i), K(\mathbf{x}_2, \mathbf{x}_i), \ldots K(\mathbf{x}_N, \mathbf{x}_i))$

$\sum_{i=1}^{N} \sum_{j=1}^{N} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j)$: a special regularizer $\beta^T \mathbf{K} \beta$

Coefficients beta_i in KLR often non-zero!

- KLR = linear model of β
  - with kernel as transform & kernel regularizer
- KLR = linear model of **w**
  - with embedded-in-kernel transform & L2 regularizer

SVM Performance

- Anecdotally they work very very well indeed.
  - Eg: They are once the best-known classifier on a well-studied hand-written-character recognition benchmark
  - Eg: Andrew knows several reliable people doing practical real-world work who claim that SVMs have saved them when their other favorite classifiers did poorly.
- There is a lot of excitement and religious fervor about SVMs as of 2001.
- Despite this, some practitioners (including your lecturer) are a little skeptical.

Doing multi-class classification

- SVMs can only handle two-class outputs
  - i.e. a categorical output variable with arity 2.
- What can be done?
- Answer: with output arity N, learn N SVM’s
  - SVM 1 learns “Output==1” vs “Output != 1”
  - SVM 2 learns “Output==2” vs “Output != 2”
  - :
  - SVM N learns “Output==N” vs “Output != N”
- Then to predict the output for a new input, just predict with each SVM and find out which one puts the prediction the furthest into the positive region.

References

- An excellent tutorial on VC-dimension and Support Vector Machines:

  C.J.C. Burges. **A tutorial on support vector machines for pattern recognition.** Data Mining and Knowledge Discovery, 2(2):955-974, 1998. http://citeseer.nj.nec.com/burges98tutorial.html

- The VC/SRM/SVM Bible:

  Statistical Learning Theory

  by Vladimir Vapnik, Wiley-Interscience; 1998

What You Should Know

- Linear SVMs
- The definition of a maximum margin classifier
- What QP can do for you
- How Maximum Margin can be turned into a QP problem
- How we deal with noisy (non-separable) data
- How we permit non-linear boundaries
- How SVM Kernel functions permit us to pretend we’re working with ultra-high-dimensional basis-function terms

电子科技大学

UESTC

1956