

# Large Margin Semi-supervised Learning

Junhui Wang and Xiaotong Shen

School of Statistics  
University of Minnesota

Email: `xshen@stat.umn.edu`

# Overview

---

*Semi-supervised learning*

*Large margin methodology*

*Computation and tuning*

*Examples*

*Theory*

# Semisupervised learning

---

- **Semisupervised learning** occurs in classification when a large amount of unlabeled data is available with only a small number of labeled data.
- **Data:** *Labeled:*  $(X_i, Y_i)_{i=1}^{n_l} \sim p(x, y)$ ; *Unlabeled:*  $(X_i)_{i=n_l+1}^{n_l+n_u} \sim p(x)$ .
- **Semisupervised learning** differs from a missing problem in that  $n_l \ll n_u$ .
- **Goal:** Enhance predictability by utilizing unlabeled data.
- **Difficulty:** 1)  $n_l \ll n_u$ , 2) need to use unlabeled data but not clear if classification can be improved.
- **Construction:**  $f(x) : \rightarrow \mathcal{R}^1$  through data. Classifier:  $Sign(f)$ .
- **Classification accuracy** is measured by generalization error (GE):

$$GE(f) = P(Y f(X) \leq 0) = \frac{1}{2} E(1 - Sign(Y f(X))).$$

# Spam email identification

- **Spam e-mail** involves sending nearly identical spam messages to many recipients.
- **Task:** Identify or classify an incoming e-mail as “spam” or not.
- **Data:**
  - **Automation:** Emails are easy to collect and vectorize.
  - **Human:** identify or label them as spam or not.
- **Goal:** Use existing data to train a filter with a few labeled e-mails and many unlabeled ones.

## Spam email identification: Spambase (UCI Repository)

Obs	make%	address%	...	\$ %	...	Capital_total	Spam?
1	0	0.64	...	0	...	278	1
2	0.21	0.28	...	0.18	...	1028	1
3	0	0	...	0	...	7	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
4600	0.3	0	...	0	...	78	?
4601	0	0	...	0	...	40	?

Response  $Y$ : 1: Spam; 0: not; ?, unlabeled.

Covariates  $X$ : (make%, address%,...,Capital\_total).

# Text categorization

- **Task:** Classify word documents into predetermined categories.
  - *Email:* Spam or non-spam.
  - *Webpage:* Sports, politics, weather, etc.
- **Automation:** Easy to obtain and vectorize word documents (unlabeled) by a computer program.
- **Human:** Time consuming to classify by human.
- **Example:** Webpage classification (WebKB, CMU).
- **Data:** Large sample size (# of documents), high dimensionality (# words), and large number of classes (# of categories).

# Existing methods

---

- **Distribution:**  $P_\theta(y|x) \leftarrow P(x)$ 
  - **Generative:** Model  $P(x|y)$ , e.g., Nigam et al. (98): EM for  $\hat{\theta}$ .
  - **Discriminative+Generative:** Model  $P(y|x)$ , e.g., (1) GRF (Zhu et al, 03). Estimate label values via smoothness. (2) Structure learning (Ando & Zhang, 05). Identify “pred. structure” from unlabeled data.
  - **Disadvantage:** 1) Assumption on  $P(y|x) \rightarrow P(x)$  (non-verifiable).  
2) Model distribution rather than decision boundary.
- **Margin:**
  - (1) Supervised:  $\min_f C \sum_{i=1}^{n_l} L(y_i f(x_i)) + \frac{1}{2} \|w\|^2$ .
  - (2) Transductive SVM (**TSVM**, Vapnik, 1998): With  $L$  the hinge loss,  
$$\min_{f, y_j} C_1 \sum_{i=1}^{n_l} L(y_i f(x_i)) + C_2 \sum_{j=n_l+1}^{n_l+n_u} L(y_j f(x_j)) + \frac{1}{2} \|w\|^2$$
  - **Disadvantage:** (1) Small sample size; (2) Unstable ← assumption?

# Our method: Large-margin

---

- **Assumption:** Label missing at random (?).
- **Goal:** Use minimal distributional assumption to yield better or no worse performance than its supervised counterpart.
- **Idea:**
  - 1) Use unlabeled data to yield *grouping boundary* (large sample).
  - 2) Use labeled data to determine *label assignment* for all data (small sample) and refine.
  - 3) Interplay: examine *all possible classification boundaries* via regularized cost function involving 1) and 2).
  - 4) Concept of *margins* in 1)-3).

# Our method: An illustration

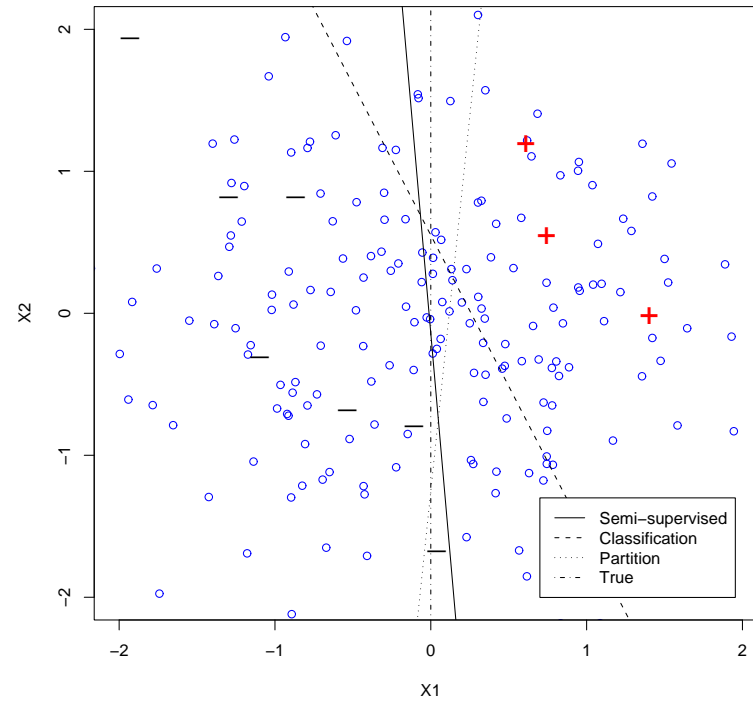


Figure 1: Illustration of our  $\psi$ -learning-based semi-supervised learning method in one example.

## Our method: Concept of margins

- Functional margins:  $z = yf(x); y_1f(x_1), \dots, y_nf(x_n)$ .
- Geometric margin: Plot.

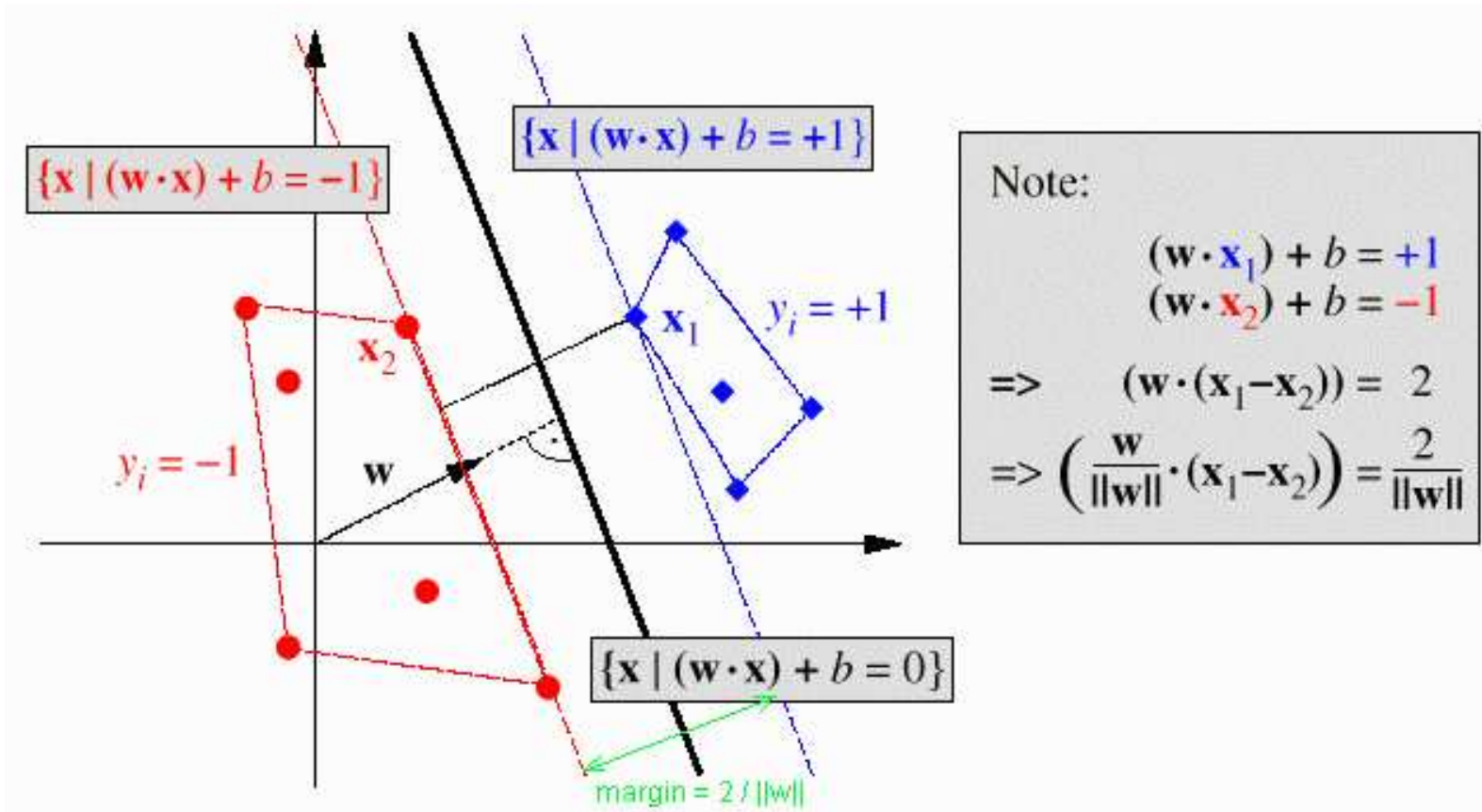


Figure 2: Illustration of geometric margin.

## Our method: Formulation

- **Classification:** Margin Loss  $L = L(yf(x))$  for decision function  $f$ .
- **Grouping:** Loss  $U = U(g(x))$  for decision function  $g$  (**Sign(g)**).
- **Key issue:** Construction  $U$  :

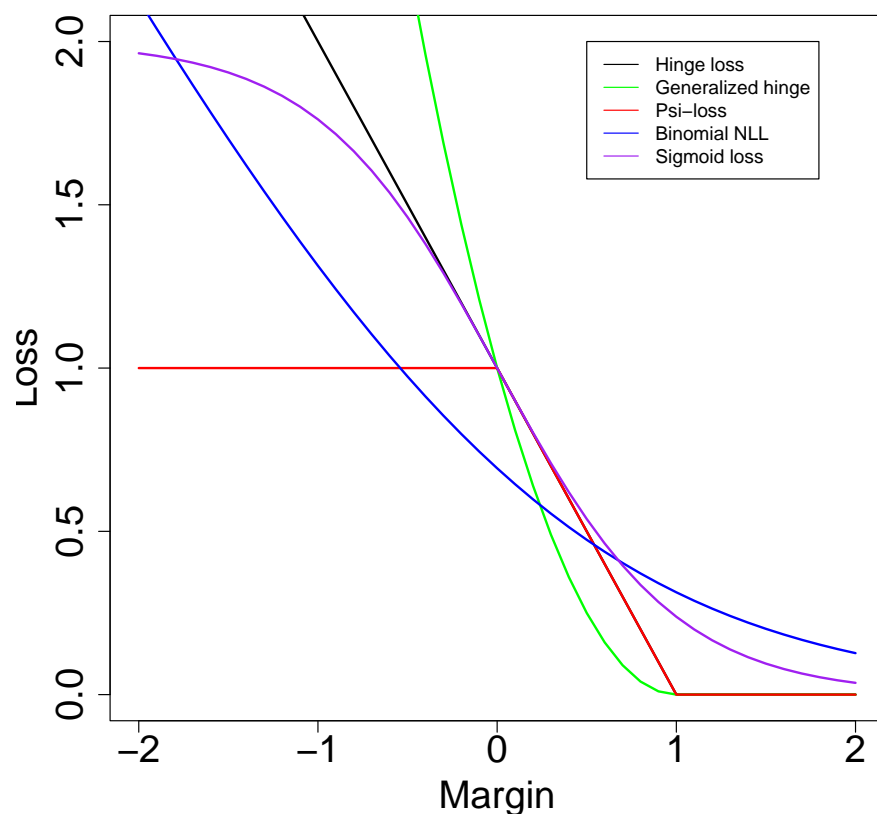
- **Regularized Loss:**

$$H(f, g) = C_1 L(yf(x)) + C_2 U(g(x)) + C_3 \|f - g\|_{p+1}^2 + \frac{1}{2} \|g\|_p^2.$$

- **Large Margin Loss:**  $L(z)$  is nonincreasing, penalizing small margin.
- **Loss for grouping:** Theorem:  $U(z) = \min_{y=\pm 1} L(yz) = L(|z|)$ .
- **Interplay:**  $\|f - g\|_{p+1}^2$  specifies a relationship between  $f$  and  $g$ .
- **Separation:**  $C_2 L(|g(x)|) + \frac{1}{2} \|g\|_p^2$ .
- **Key:**  $\arg \inf (EU(g(x)) + \frac{1}{2} \|g\|_p^2)$  approximates the Bayes rule.

## Our method: Margin loss $L$

Misclassification loss is intractable. Various margin losses have been proposed. A margin approach is expected to perform better than classification via regression.



- Hinge loss (Vapnik, 1995):  
 $L(z) = (1 - z)_+ = \max(1 - z, 0)$ .
- Generalized hinge loss (Lin, 2002):  
 $L(z) = (1 - z)_+^q; q > 1$ .
- $\psi$ -loss (Shen, Tseng, Zhang and Wong, 2003):  
 $L(z) = \min(1, (1 - z)_+)$ .
- Logistic loss (Zhu and Hastie, 2004):  
 $L(z) = \log(1 + e^{-z})$
- Sigmoid loss (Mason et al., 2000):  
 $L(z) = 1 - \tanh(\lambda z)$ .

Figure 3: Plot of various losses versus margin.

# Our method: Framework

- Cost function

- *Linear*:  $f, g$  (Linear representation):  $b + \sum_{i=1}^p \alpha_i x_i$

$$h = C_1 \sum_{i=1}^{n_l} L(y_i f(x_i)) + C_2 \sum_{j=n_l+1}^{n_l+n_u} U(g(x_j)) + \frac{C_3}{2} \|f - g\|_{p+1}^2 + \frac{1}{2} \|g\|_p^2.$$

- *Nonlinear*:  $f, g$  (RKHS Representation):  $b + \sum_{i=1}^{n_l+n_u} \alpha_i K(x, x_i)$ .

$$h_K = C_1 \sum_{i=1}^{n_l} L(y_i f(x_i)) + C_2 \sum_{j=n_l+1}^{n_l+n_u} U(g(x_j)) + \frac{C_3}{2} \|f - g\|_{p+1, K}^2 + \frac{1}{2} \|g\|_{p, K}^2.$$

- Minimization of  $h(f, g)$  yields  $(\hat{f}, \hat{g})$  thus classifier  $Sign(\hat{f})$ .

# Our method: Nonconvex Minimization

- *Difficulty*:  $L$  is convex or nonconvex;  $U$  is non-convex.
- *Difference Convex Algorithm* (An & Tao, 1997, Liu, Shen & Wong, 2005).
  - Hinge loss:  $U = (1 - |z|)_+ = (|z| - 1)_+ - (|z| - 1)$ .
  - $\psi$ -loss:  $\psi = 2(1 - z)_+ - 2(-z)_+$ ;  $U(z) = 2(1 - |z|)_+$ .

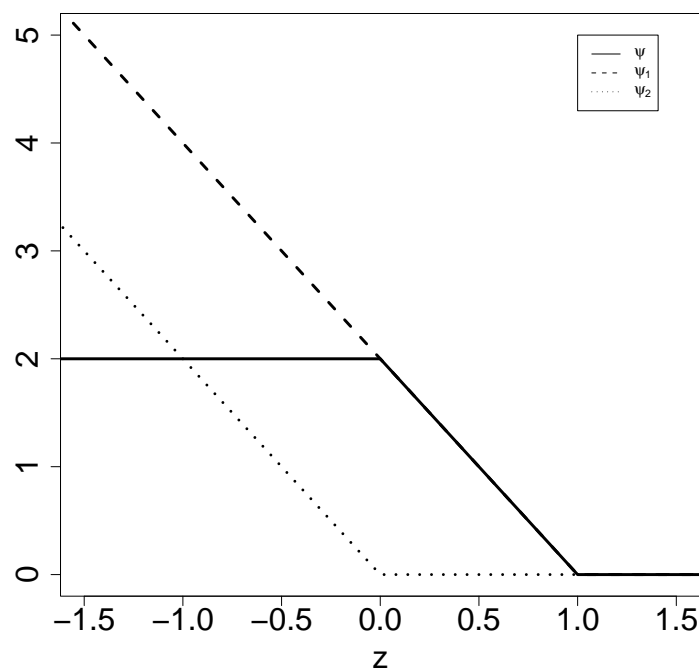
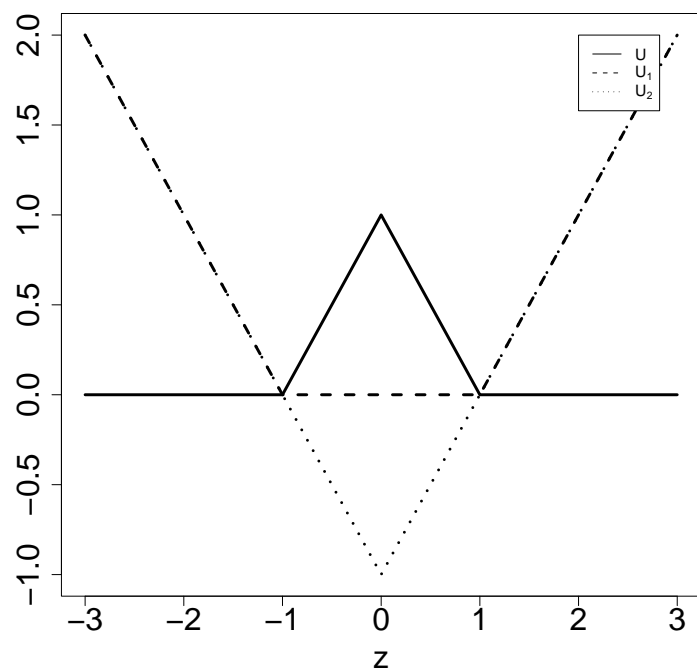


Figure 4: DC decomposition.

## Our method: Computation

- Key:  $h(f, g) = h_1(f, g) - h_2(f, g)$ , with  $h_1$  and  $h_2$  convex.

*Algorithm: (Sequential quadratic programming)*

**Step 1:** Initialize  $f^{(0)} = g^{(0)}$  with SVM on labeled data alone.

**Step 2:** At iteration  $k + 1$ , solve

$$\min_{f, g} h_1(f, g) - \langle (f, g), \nabla h_2(f^{(k)}, g^{(k)}) \rangle,$$

with  $\nabla h_2(f, g)$  a gradient vector of  $h_2$ .

**Step 3:** Terminate when  $|h(f^{(k+1)}, g^{(k+1)}) - h(f^{(k)}, g^{(k)})| \leq \epsilon$ .

- *Theorem:* (Convergence)  $h(f^{(k)}, g^{(k)})$  is nonincreasing,  $\lim_{k \rightarrow \infty} h(f^{(k)}, g^{(k)}) \geq \min_{(f, g)} h(f, g)$ , and  $\lim_{k \rightarrow \infty} \|(f^{(k+1)}, g^{(k+1)}) - (f^{(k)}, g^{(k)})\| = 0$ . Moreover, the Algorithm terminates finitely.

- Convergence  $\rightarrow$  *superlinear*. Complexity  $\rightarrow$  QP.

# Our method: Computation, continued

- Model tuning

- Performance needs to be optimized wrt  $C = (C_1, C_2, C_3)$ .

- Estimation of  $GE(\hat{f}_C)$ .

$$\widehat{GE}(\hat{f}_C) = EGE(\hat{f}_C) + \sum_{i=1}^{n_l} Cov(Y_i, Sign(\hat{f}_C(X_i))) + Correction.$$

- *Data perturbation* and MC approximation, c.f, Shen and Huang (2005); Wang and Shen (2006).

- Better estimation scheme is under investigation to use unlabeled data more effectively.

# Numerical Examples

---

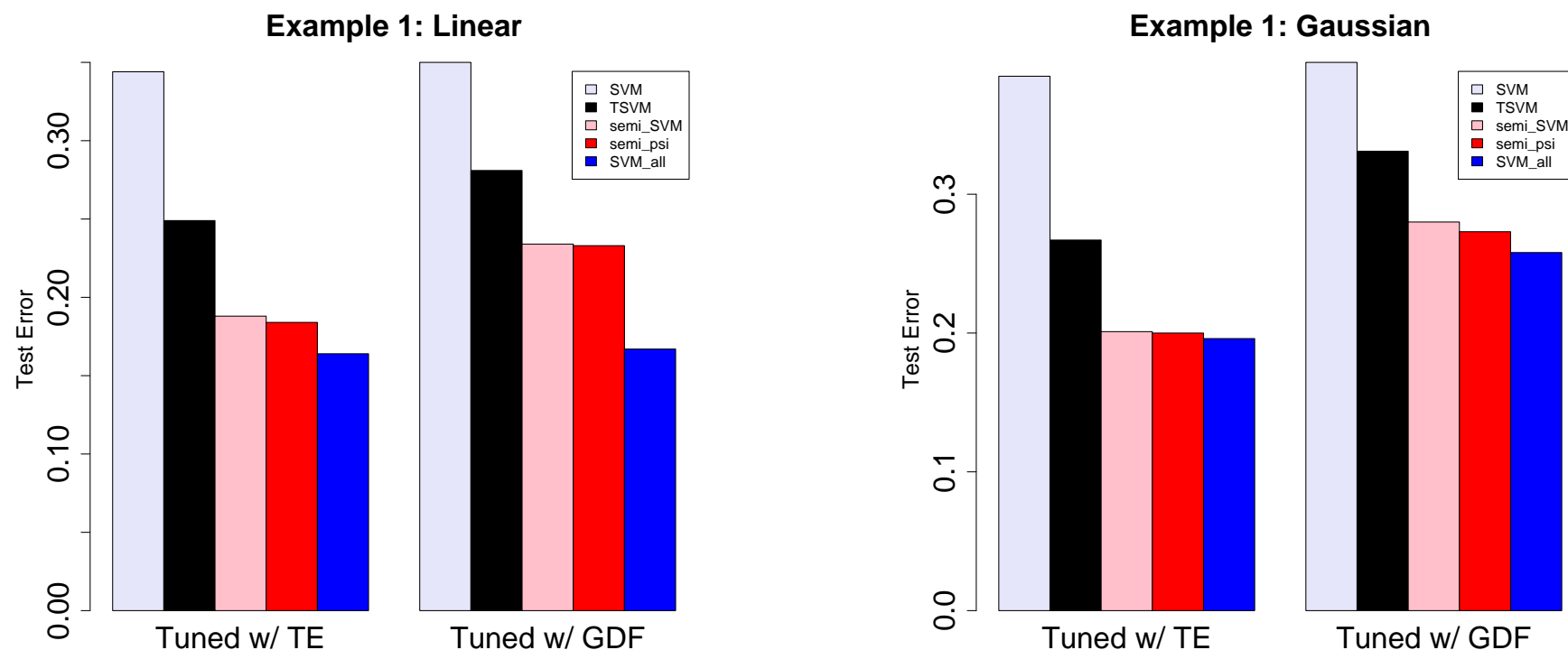
- **Example 1:**  $(Y_i, X_{i1}, X_{i2})_{i=1}^{1000}$  are sampled i.i.d. according to  $(Y_i + 1)/2 \sim \text{Bern}(0.5)$ ,  $X_{i1} \sim N(Y_i, 1)$ , and  $X_{i2} \sim N(0, 1)$ .
- **Example 2:**  $(Y_i, X_{i1}, X_{i2})_{i=1}^{1000}$  are sampled i.i.d. according to  $X_{i1} \sim N(3 \cos(k_i \pi / 2 + \pi / 8), 1)$ ,  $X_{i2} \sim N(3 \sin(k_i \pi / 2 + \pi / 8), 4)$  with  $k_i$  sampled uniformly from  $\{1, 2, 3, 4\}$ , and  $Y_i = 1$  if  $k_i \in \{1, 4\}$  and  $-1$  o.w.
- Three benchmark examples: **WBC, mushroom and spam email**.
- **Performances** are evaluated by averaged test errors of 100 random replicates.

## Numerical Examples, Continued

<i>Data</i>	dimension	labeled size	unlabeled size	testing size
Example1	$1000 \times 2$	10	190	800
Example2	$1000 \times 2$	10	190	800
WBC	$682 \times 9$	10	190	482
Mushroom	$8124 \times 22$	10	190	7924
Email	$4601 \times 57$	20	580	4001

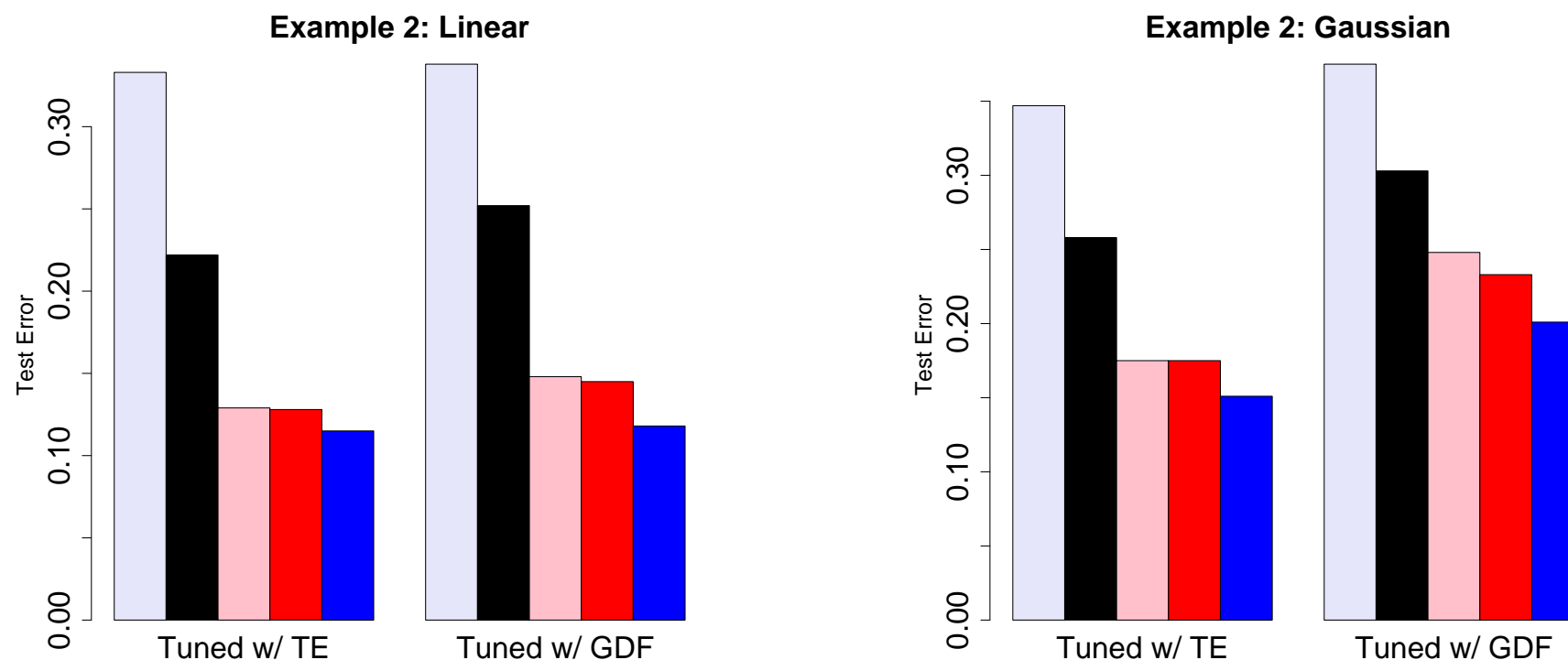
Table 1: Description of all data examples in our simulations.

# Generalization error: Example 1



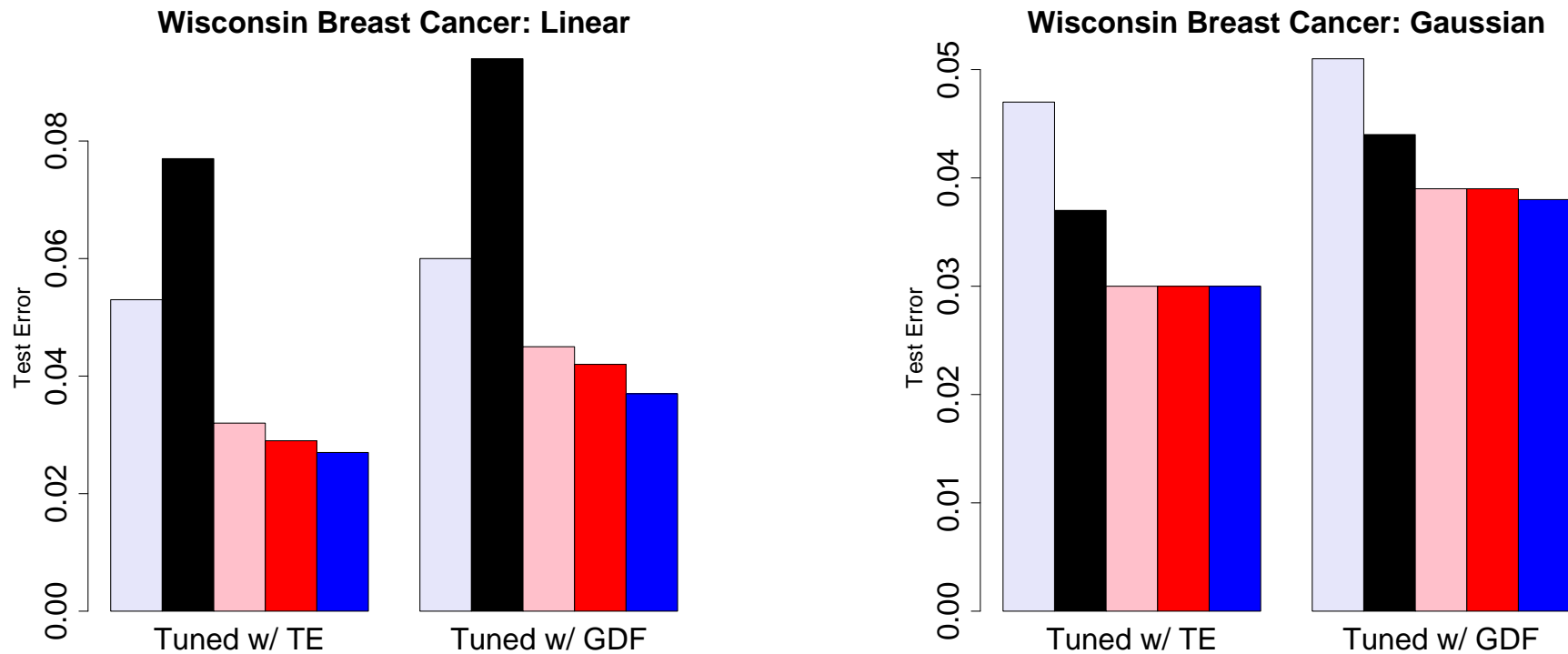
- Our methods outperform *SVM* with labeled data alone and *TSVM*.
- $\psi$ -loss performs slightly better than *hinge loss*.
- Performance of our methods is close to *SVM* without missing labels.

## Generalization error: Example 2



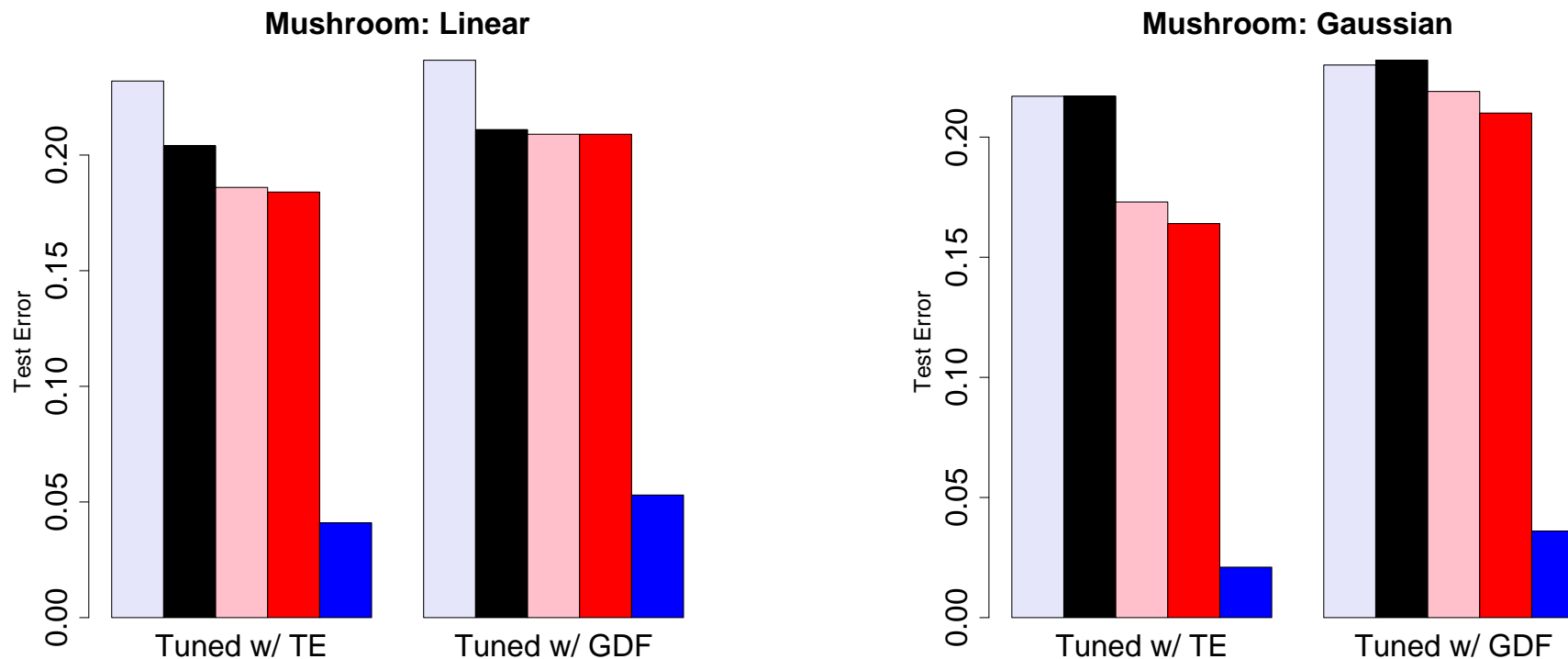
- Our methods outperform *SVM* with labeled data alone and *TSVM*.
- $\psi$ -loss performs slightly better than *hinge loss*.
- Performance of our methods is close to *SVM* without missing labels.

# Generalization error: Wisconsin Breast Cancer



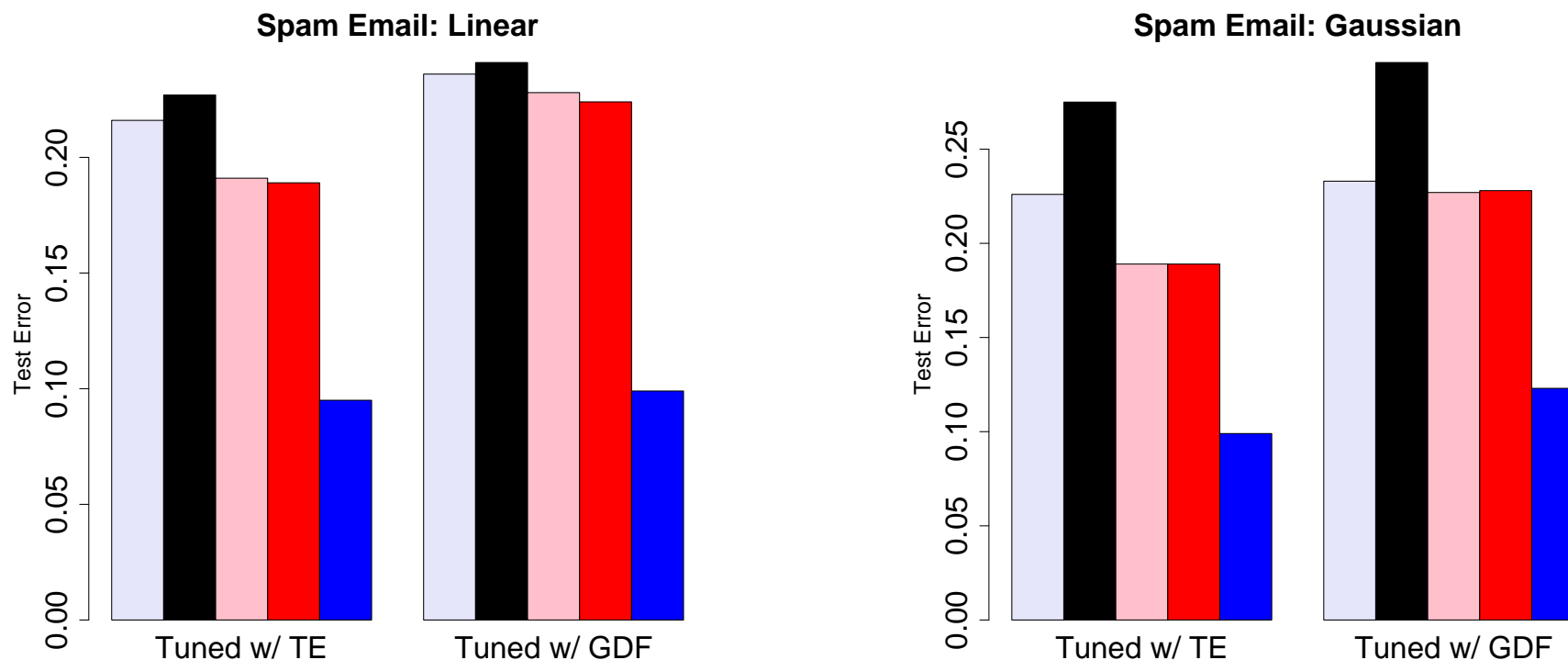
- Our methods outperform *SVM* with labeled data alone and *TSVM*.
- $\psi$ -loss performs slightly better than hinge loss.
- Performance of our methods is close to *SVM* without missing labels.
- *TSVM* performs worse than *SVM* with labeled data alone in the linear case.

# Generalization error: Mushroom



- Our methods outperform *SVM* with labeled data alone and *TSVM*.
- $\psi$ -loss performs slightly better than *hinge loss*.
- Performance of our methods is close to *SVM* without missing labels.
- *TSVM* performs worse than *SVM with labeled data alone* in the Gaussian case.

# Generalization error: Spam Email



- Our methods outperform *SVM with labeled data alone* and *TSVM*.
- *$\psi$ -loss* performs slightly better than *hinge loss*.
- Performance of our methods is close to *SVM without missing labels*.
- *TSVM* performs worse than *SVM* with labeled data alone in both the linear and Gaussian cases.

# Learning Theory: $\psi$ -learning

---

- *Performance:*

- **Classification:**  $e(f, f^*) = GE(f) - GE(f^*) \geq 0$  with  $f^* = \arg \inf_{f \in \mathcal{F}} EL(Y f(X))$ .
- **Grouping:**  $e_{\tilde{U}}(g, g_C^*) = E\tilde{U}(g) - E\tilde{U}(g_C^*)$  with  $\tilde{U}(g) = U(g) + \frac{1}{C_2} J(f_C^*, g)$ ,  $J(f, g) = \frac{C_3}{2} \|g - f\|_{p+1, K}^2 + \frac{1}{2} \|g\|_{p, K}^2$ , and  $(f_C^*, g_C^*)$  is the minimizer of  $EH(f, g)$

- *Penalty:*

$$J_0 = \max(J(f_C^*, g_C^*), 1), \quad J_0^L = \max\left(\frac{C_3}{4} \|f_C^* - g_C^*\|_{p+1, K}^2, 1\right),$$
$$J_0^U = \max\left(\frac{C_3}{4} \|f_C^* - g_C^*\|_{p+1, K}^2 + \frac{1}{2} \|g_C^*\|_{p, K}^2, 1\right).$$

# Learning Theory: *Assumptions*

- (A). (**Smoothness**) For constants  $0 < \alpha < \infty$ ,  $0 \leq \beta < 2$ ,  $a_1 > 0$ ,  $a_2 > 0$  and all small  $\delta > 0$ ,

$$\sup_{\{g \in \mathcal{F}: e_{\tilde{U}}(g, g_C^*) \leq \delta\}} |e(g, g_C^*)| \leq a_1 \delta^\alpha$$

$$\sup_{\{g \in \mathcal{F}: e_{\tilde{U}}(g, g_C^*) \leq \delta\}} \text{Var}(U(g) - U(g_C^*)) \leq a_2 \delta^\beta$$

- (B). (**Metric entropy**) For  $\epsilon_{n_u} > 0$ ,  $\sup_{k \geq 2} \phi(\epsilon_{n_u}, k) \leq a_5 n_u^{1/2}$ , where

$$\phi(\epsilon, k) = \int_{a_4 T}^{a_3^{1/2} T^{1/2}} H^{1/2}(w^2/4, \mathcal{G}(k)) dw / T, \text{ with}$$

$$T = T(\epsilon, C, k) = \min(\epsilon/2 + (n_u C_2)^{-1} (k/2 - 1) J_0, 1).$$

- (C). (**Norm relationship**) For constant  $a_6 > 0$  and any  $f \in \mathcal{F}$ ,  $\|f\|_2^2 \leq a_6 \|f\|_{p+1, K}^2$  with  $\|\cdot\|_2$  the  $L_2$  norm.

# Learning Theory: Main results

**Theorem:** ( $\psi$ -learning) Under Assumptions A-C, there exist constants  $a_7, a_8, a_9 > 0$  such that

$$\begin{aligned} & P \left( \inf_C |e(\hat{f}_C, f^*)| \geq a_1 (2\delta_{n_u}^2)^\alpha + \inf_C |e(g_C^*, f^*)| \right) \\ & \leq 6.5 \exp(-a_8 n_l (n_l C_1)^{-1} J_0^L) + 6.5 \exp(-a_9 n_u (n_u C_2)^{-1} J_0^U) + \\ & \quad 3.5 \exp(-a_7 n_u ((n_u C_2)^{-1} J_0)^{\max(1, 2-\beta)}), \end{aligned}$$

where  $\delta_{n_u}^2 = \min(\epsilon_{n_u}^{\frac{2}{2-\beta}}, 1)$ .

**Corollary:** As  $n_u \geq n_l \rightarrow \infty$ ,

$$\inf_C |e(\hat{f}_C, f^*)| = O_p \left( \min \left( \delta_{n_l}^{2\alpha} \inf_C |e(g_C^*, f^*)| + \delta_{n_u}^{2\alpha} \right) \right),$$

**Conclusion:** Achieve the objective. If approximation is adequate, then  $O_p(\delta_{n_u}^{2\alpha})$  (**Better**); otherwise  $O_p(\delta_{n_l}^{2\alpha})$  (**No worse**).

# Linear Example

- **Data:**  $(Y, X_{(1)}, X_{(2)}), X_{(k)}$ ;  $k = 1, 2$  sampled i.i.d. from  $q(x) = \frac{1}{2}(\theta + 1)|x|^\theta$  for  $|x| \leq 1$ ,  $Y$  sampled i.i.d. according to  $P(Y = 1|X_{(1)} = x) = 1 - \tau$  if  $x \geq 0$  and  $\tau$  o.w. with  $0 < \tau < 1$ .
- **Approximation error:**  $\inf_C |e(g_C^*, f^*)| = 0$ .
- **Error Rate:**  $\inf_C |e(\hat{f}_C, f^*)| = O_p((n_u^{-1} \log n_u)^{(\theta+1)/2})$ .
- **Remarks:**
  - Convergence is *fast* due to  $n_u \gg n_l$ .
  - Convergence can be *arbitrarily fast* as  $\theta \rightarrow \infty$ .
  - Here  $n_l$  appears to have *little* contribution to GE of  $\hat{f}_C$ .

# Take Away Messages

---

- Future work:
  - More efficient loss for grouping.
  - Regularization solution path.
  - Multi-class and large margin semisupervised clustering.
  - Active learning. Select “most informative” unlabeled covariates to obtain additional labels to improve accuracy of classification.
  - Variable selection, Inference, Prediction within semisupervised learning, ...