

The Adviceptron: Giving Advice To The Perceptron

Gautam Kunapuli¹, Kristin P. Bennett²,
Richard Maclin³ and Jude W. Shavlik¹

¹University of Wisconsin-Madison, USA

²Rensselaer Polytechnic Institute, USA

³University of Minnesota, Duluth, USA

Outline

- ***Knowledge-Based Support Vector Machines***
- The Adviceptron: Online KBSVMs
- A Real-World Task: Diabetes Diagnosis
- A Real-World Task: Digit Recognition
- Conclusions

Knowledge-Based SVMs

- Introduced by Fung et al (2003)
- Allows incorporation of expert advice into SVM formulations
- Advice is specified with respect to polyhedral regions in input (feature) space

$$(\text{feature}_7 \geq 5) \wedge (\text{feature}_{12} \leq 4) \Rightarrow (\text{class} = +1)$$

$$(\text{feature}_2 \leq -3) \wedge (\text{feature}_3 \leq 4) \wedge (\text{feature}_{10} \geq 0) \Rightarrow (\text{class} = -1)$$

$$(3\text{feature}_6 + 5\text{feature}_8 \geq 2) \wedge (\text{feature}_{11} \leq -3) \Rightarrow (\text{class} = +1)$$

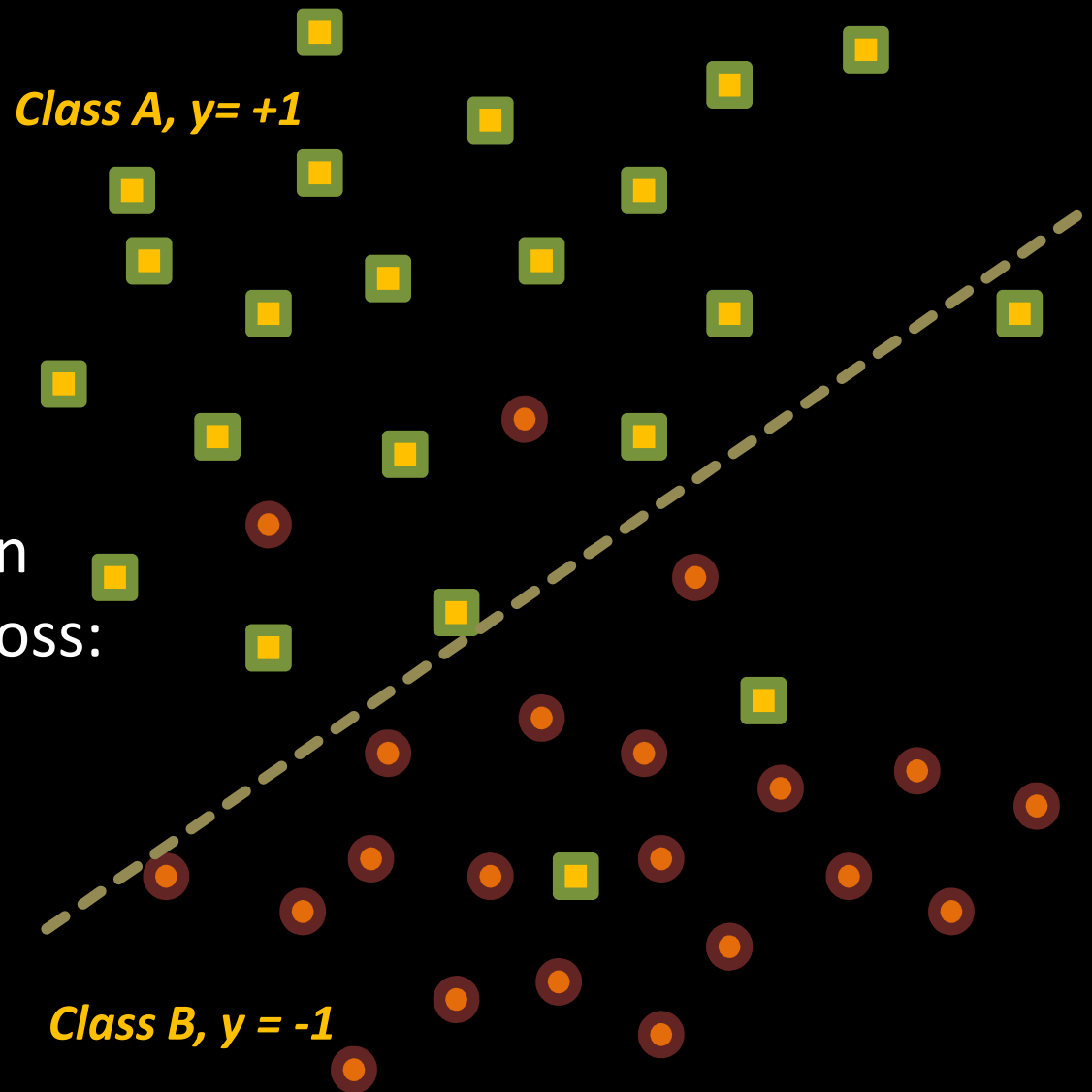
- Can be incorporated into SVM formulation as constraints using ***advice variables***

Knowledge-Based SVMs

In classic SVMs, we have T labeled data points (\mathbf{x}^t, y_t) , $t = 1, \dots, T$. We learn a linear classifier $\mathbf{w}'\mathbf{x} - b = 0$.

The standard SVM formulation trades off regularization and loss:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \lambda \mathbf{e}'\boldsymbol{\xi} \\ \text{sub. to} \quad & Y(X\mathbf{w} - b\mathbf{e}) + \boldsymbol{\xi} \geq \mathbf{e}, \\ & \boldsymbol{\xi} \geq \mathbf{0}. \end{aligned}$$



Knowledge-Based SVMs

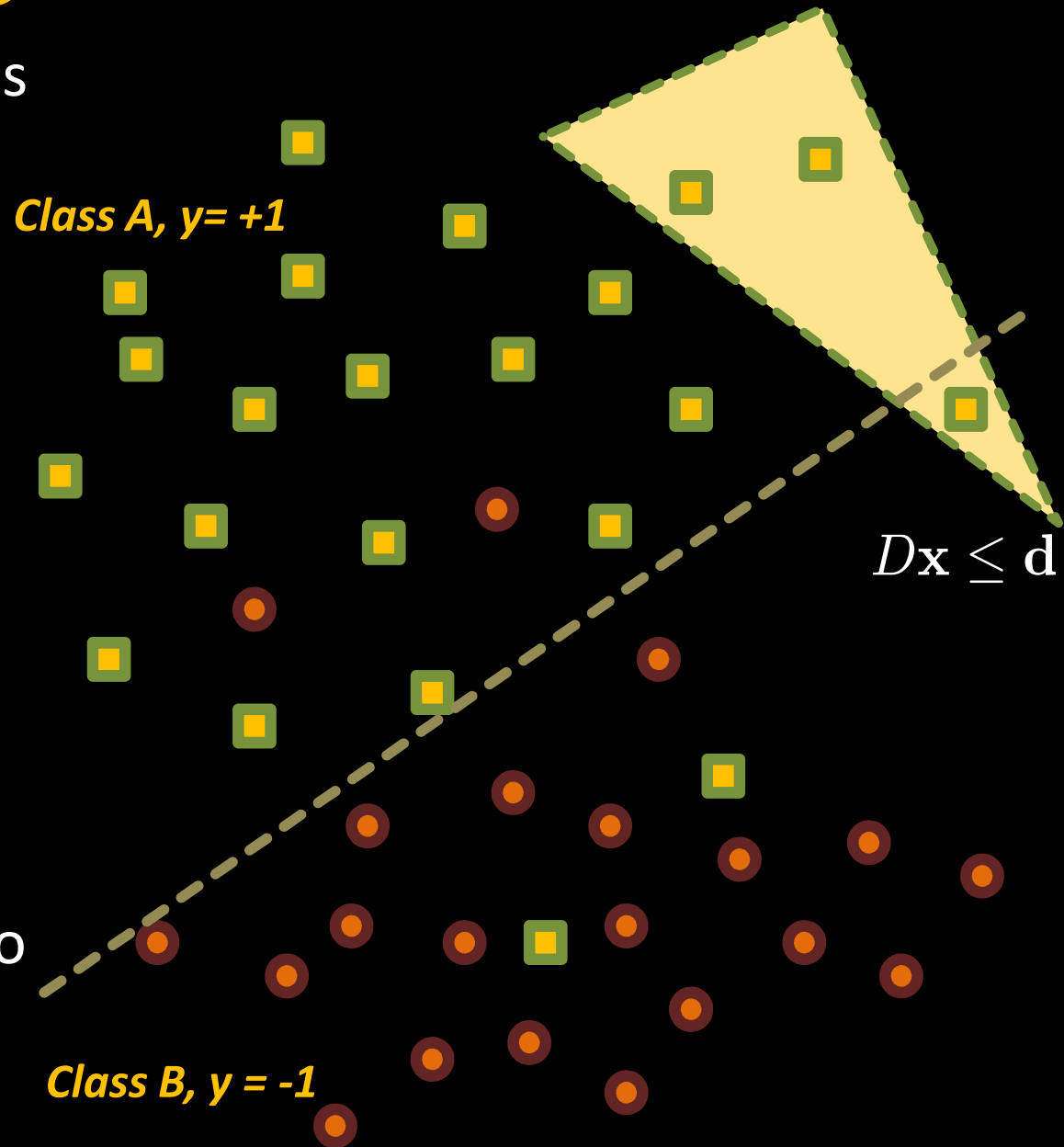
We assume an expert provides **polyhedral advice** of the form

$$D\mathbf{x} \leq \mathbf{d} \Rightarrow \mathbf{w}'\mathbf{x} \geq b$$

We can transform the logic constraint above using **advice variables, \mathbf{u}**

$$\begin{aligned} D'\mathbf{u} + \mathbf{w} &= 0, \\ -\mathbf{d}'\mathbf{u} - b &\geq 0, \\ \mathbf{u} &\geq 0 \end{aligned}$$

These constraints are added to the standard formulation to give **Knowledge-Based SVMs**



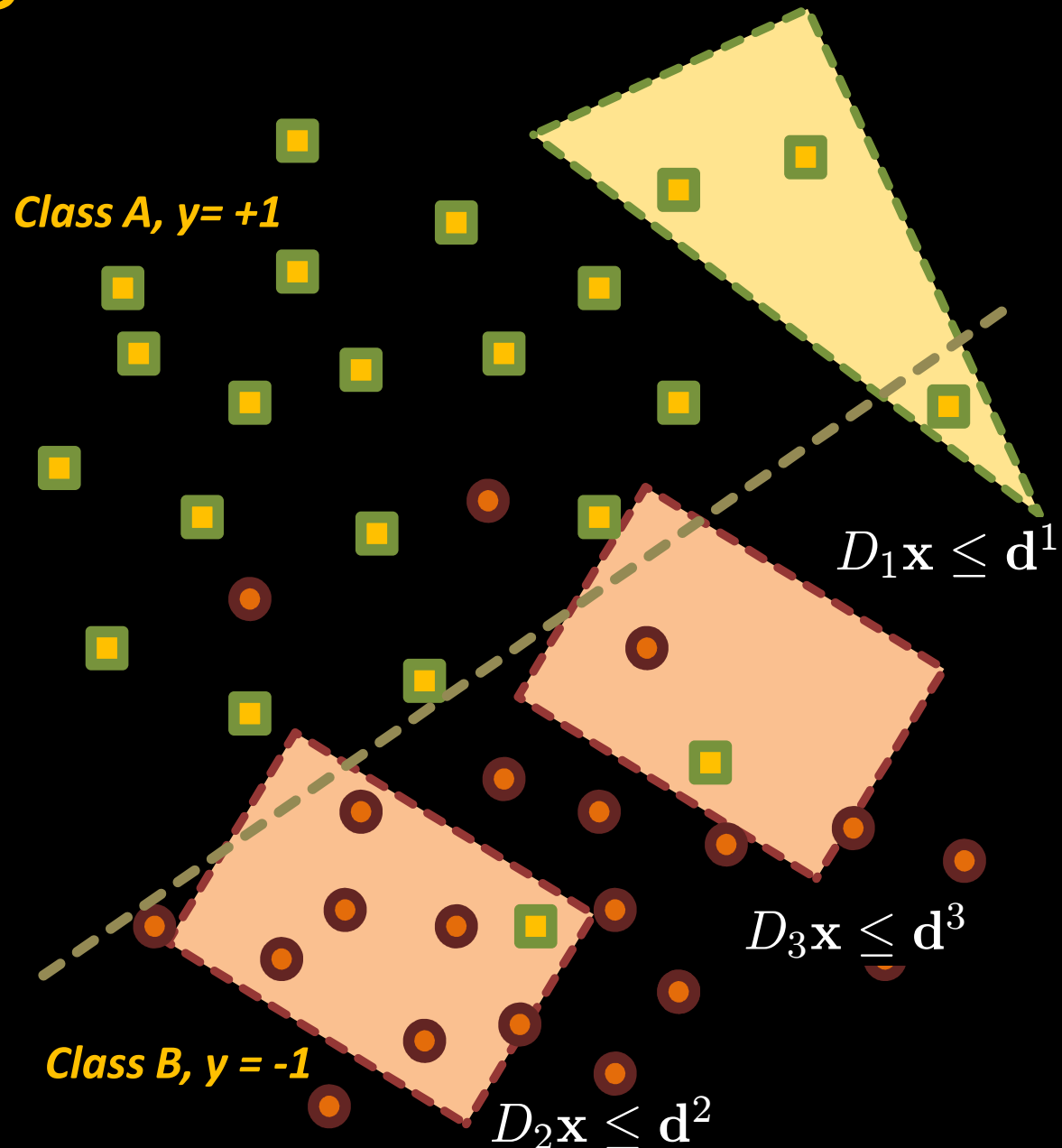
Knowledge-Based SVMs

In general, there are **m advice sets**, each with label $z = \pm 1$ for advice belonging to Class A or B,

$$D_i \mathbf{x} \leq \mathbf{d}^i \Rightarrow z_i (\mathbf{w}' \mathbf{x}) - b \geq 0$$

Each advice set **adds the following constraints** to the SVM formulation

$$\begin{aligned} D_i' \mathbf{u}^i + z_i \mathbf{w} &= 0, \\ -\mathbf{d}^{i'} \mathbf{u}^i - z_i b &\geq 0, \\ \mathbf{u}^i &\geq 0 \end{aligned}$$



Knowledge-Based SVMs

The batch KBSVM formulation introduces **advice slack variables** to **soften** the advice constraints

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + \lambda \mathbf{e}'\xi + \mu \sum_{i=1}^m (\eta^i + \zeta_i)$$

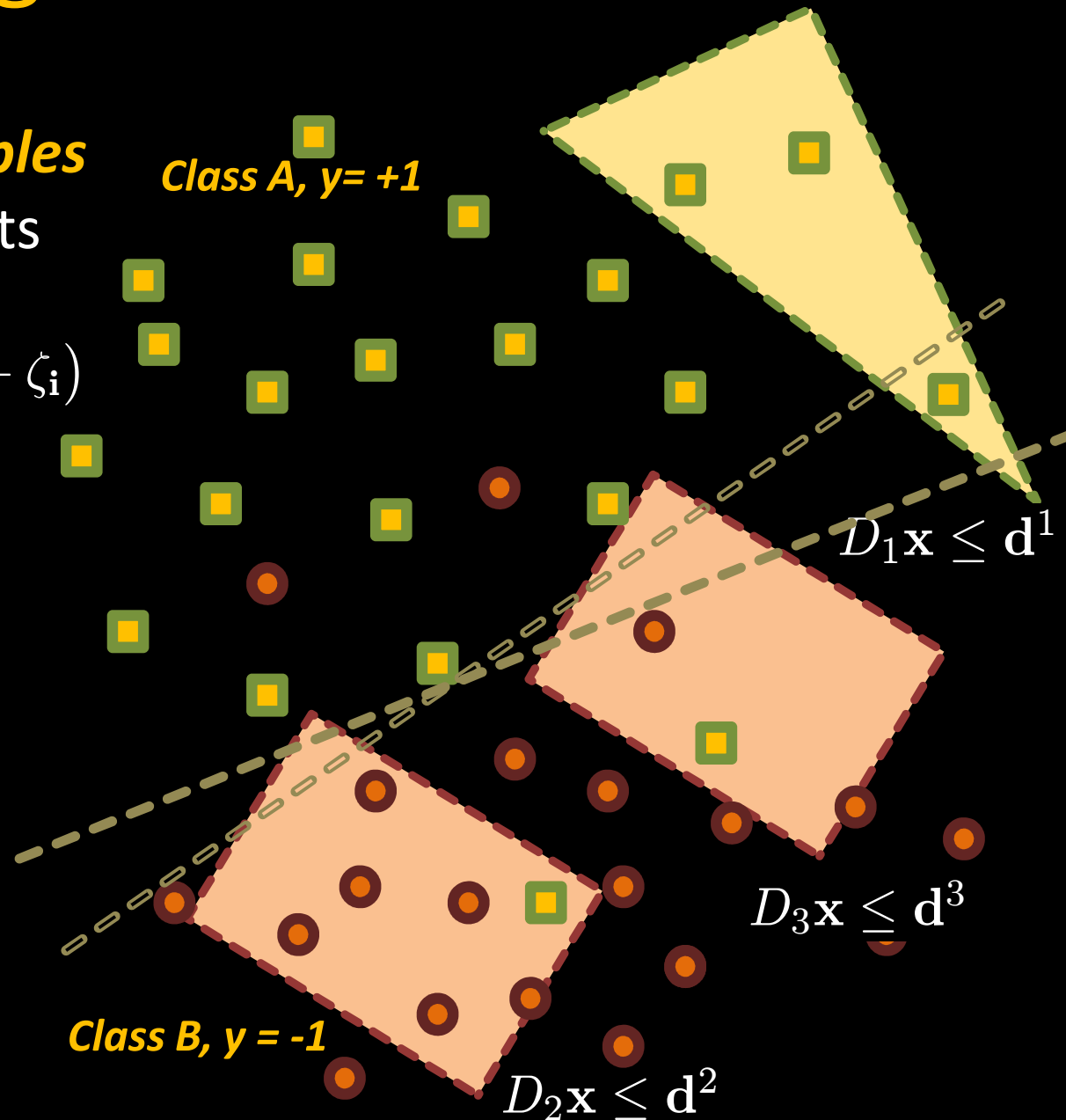
$$\text{s.t.} \quad Y(X\mathbf{w} - b\mathbf{e}) + \xi \geq \mathbf{e},$$

$$\xi \geq 0,$$

$$D'_i \mathbf{u}^i + z_i \mathbf{w} + \eta^i = 0,$$

$$-\mathbf{d}^{i'} \mathbf{u}^i - z_i b + \zeta_i \geq 1,$$

$$\mathbf{u}^i, \eta^i, \zeta_i \geq 0, \quad i = 1, \dots, m.$$



Outline

- Knowledge-Based Support Vector Machines
- ***The Adviceptron: Online KBSVMs***
- A Real-World Task: Diabetes Diagnosis
- A Real-World Task: Digit Recognition
- Conclusions

Online KBSVMs

- Need to derive an *online version of KBSVMs*
- Algorithm is provided with advice and *one* labeled data point at each round
- Algorithm should *update the hypothesis* at each step, w^t , *given fixed advice vectors*, $u^{i,*}$

Learning From Knowledge Only

- We assume that **advice vectors** $\mathbf{u}^{i,*}$ are **pre-computed** based on the expert advice **before** any data are available.
- Can do this by solving **advice-only KBSVM**

$$\begin{aligned} & \min_{\mathbf{u}^i \geq 0, \mathbf{w}, \boldsymbol{\eta}^i, \zeta_i} \frac{1}{2} \|\mathbf{w}\|_2^2 + \mu \sum_{i=1}^m \left(\|\boldsymbol{\eta}^i\|_2^2 + \zeta_i^2 \right) \\ & \text{subject to } D'_i \mathbf{u}^i + z_i \mathbf{w} + \boldsymbol{\eta}^i = 0, \\ & \quad -\mathbf{d}^{i'} \mathbf{u}^i - z_i b + \zeta_i \geq 0, \quad i = 1, \dots, m. \end{aligned}$$

Deriving the Advicetreron

- There are m advice vectors, $\mathbf{u}^{i,*}$, $i = 1, \dots, m$
- At round t , the algorithm receives (\mathbf{x}^t, y_t)
- The **current hypothesis** is \mathbf{w}^t

Define $\sigma_t^h = \begin{cases} 1, & \text{if } y_t \mathbf{w}^{t'} \mathbf{x}^t \leq 0, & \text{(misclassification)} \\ 0, & \text{if } y_t \mathbf{w}^{t'} \mathbf{x}^t > 0, & \text{(correct classification)} \end{cases}$


$\sigma_t^h = 1$

- In classical perceptron, updates $c_{\sigma_t^h} = 0$
- In advicetreron, updates even for $\sigma_t^h = 1$ based on **advice**

Advice Updates When $\sigma_t^h = 0$

- There are m advice vectors, $\mathbf{u}^{i,\star}$, $i = 1, \dots, m$
- At round t , the algorithm receives (\mathbf{x}^t, y_t)
- The **current hypothesis** is \mathbf{w}^t


proximal term for hypothesis, keeps update as close to current hypothesis as possible


$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w} - \mathbf{w}^t\|_2^2 + \frac{\mu}{2} \sum_{i=1}^m \|\boldsymbol{\eta}^i\|_2^2, \\ \text{s. t.} \quad & D_i' \mathbf{u}^{i,\star} + z_i \mathbf{w} + \boldsymbol{\eta}^i = 0, \quad i = 1, \dots, m. \end{aligned}$$

Advice Updates When $\sigma_t^h = 0$

- There are m advice vectors, $\mathbf{u}^{i,\star}$, $i = 1, \dots, m$
- At round t , the algorithm receives (\mathbf{x}^t, y_t)
- The **current hypothesis** is \mathbf{w}^t

no data loss as \mathbf{w}^t classifies \mathbf{x}^t correctly
objective minimizes advice loss only;

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w} - \mathbf{w}^t\|_2^2 + \frac{\mu}{2} \sum_{i=1}^m \|\boldsymbol{\eta}^i\|_2^2, \\ \text{s. t.} \quad & D_i' \mathbf{u}^{i,\star} + z_i \mathbf{w} + \boldsymbol{\eta}^i = 0, \quad i = 1, \dots, m. \end{aligned}$$


Advice Updates When $\sigma_t^h = 0$

- There are m advice vectors, $\mathbf{u}^{i,\star}$, $i = 1, \dots, m$
- At round t , the algorithm receives (\mathbf{x}^t, y_t)
- The **current hypothesis** is \mathbf{w}^t

Equality constraint measures loss with respect to advice; error variables can be eliminated

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w} - \mathbf{w}^t\|_2^2 + \frac{\mu}{2} \sum_{i=1}^m \|\boldsymbol{\eta}^i\|_2^2, \\ \text{s. t.} \quad & D_i' \mathbf{u}^{i,\star} + z_i \mathbf{w} + \boldsymbol{\eta}^i = 0, \quad i = 1, \dots, m. \end{aligned}$$

Advice Updates When $\sigma_t^h = 0$

- There are m advice vectors, $\mathbf{u}^{i,*}$, $i = 1, \dots, m$
- At round t , the algorithm receives (\mathbf{x}^t, y_t)
- The current hypothesis is \mathbf{w}^t

fixed, *advice-estimate of the hypothesis according to i -th advice set*; denote as $\mathbf{r}^{i,t}$

parameter controls the influence of advice

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w} - \mathbf{w}^t\|_2^2 + \frac{\mu}{2} \sum_{i=1}^m \|\boldsymbol{\eta}^i\|_2^2, \\ \text{s. t.} \quad & \mathbf{D}'_i \mathbf{u}^{i,*} + z_i \mathbf{w} + \boldsymbol{\eta}^i = 0, \quad i = 1, \dots, m. \end{aligned}$$

Advice Updates When $\sigma_t^h = 0$

- There are m advice vectors, $\mathbf{u}^{i,*}$, $i = 1, \dots, m$
- At round t , the algorithm receives (\mathbf{x}^t, y_t)
- The current hypothesis is \mathbf{w}^t

fixed, *advice-estimate of the hypothesis according to i -th advice set*; denote as $\mathbf{r}^{i,*}$

average *advice-estimates over all m advice vectors* and denote as

$$\mathbf{r}^* = \frac{1}{m} \sum_{i=1}^m \mathbf{r}^{i,*}$$

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w} - \mathbf{w}^t\|_2^2 + \frac{\mu}{2} \sum_{i=1}^m \|\boldsymbol{\eta}^i\|_2^2,$$

$$\text{s. t. } D_i' \mathbf{u}^{i,*} + z_i \mathbf{w} + \boldsymbol{\eta}^i = 0, \quad i = 1, \dots, m.$$

Advice Updates When $\sigma_t^h = 0$

Given advice-estimate \mathbf{r}^* , the **update** when the **current hypothesis does *not*** make a mistake is

$$\mathbf{w}^{t+1} = \nu \mathbf{w}^t + (1 - \nu) \mathbf{r}^*$$

Advice Updates When $\sigma_t^h = 0$

Given advice-estimate r^* , the **update** when the **current hypothesis does *not*** make a mistake is

$$\mathbf{w}^{t+1} = \nu \mathbf{w}^t + (1 - \nu) \mathbf{r}^*$$

Update is **convex combination** of the **current hypothesis** and the **average advice-estimate**

Parameter of convex combinations is $\nu = \frac{1}{1 + m\mu}$

Advice Updates When $\sigma_t^h = 0$

Given advice-estimate \mathbf{r}^* , the **update** when the **current hypothesis does *not* make a mistake** is

$$\mathbf{w}^{t+1} = \nu \mathbf{w}^t + (1 - \nu) \mathbf{r}^*$$

The **margin** of the updated hypothesis is

$$\gamma = \nu y_t \mathbf{w}^{t'} \mathbf{x}^t + (1 - \nu) y_t \mathbf{r}^{*'} \mathbf{x}^t$$

Update is **convex combination** of the **current hypothesis** and the **average advice-estimate**

Parameter of convex combinations is $\nu = \frac{1}{1 + m\mu}$

Advice Updates When $\sigma_t^h = 0$

Given advice-estimate r^* , the **update** when the **current hypothesis does *not* make a mistake** is

$$\mathbf{w}^{t+1} = \nu \mathbf{w}^t + (1 - \nu) \mathbf{r}^*$$

The **margin** of the updated hypothesis is

$$\gamma = \nu y_t \mathbf{w}^{t'} \mathbf{x}^t + (1 - \nu) y_t \mathbf{r}^{*'} \mathbf{x}^t$$

margin is **convex combination** of margin according to **current hypothesis** and the **average advice-estimate**

Update is **convex combination** of the **current hypothesis** and the **average advice-estimate**

Parameter of convex combinations is $\nu = \frac{1}{1 + m\mu}$

Advice Updates When $\sigma_t^h = 0$

Given advice-estimate \mathbf{r}^* , the **update** when the **current hypothesis does *not* make a mistake** is

$$\mathbf{w}^{t+1} = \nu \mathbf{w}^t + (1 - \nu) \mathbf{r}^*$$

The **margin** of the updated hypothesis is

$$\gamma = \nu y_t \mathbf{w}^{t'} \mathbf{x}^t + (1 - \nu) y_t \mathbf{r}^{*'} \mathbf{x}^t$$

Use this margin to determine **if there is an advice update**

if $\gamma \leq 0$, update according to $\mathbf{w} = \nu \mathbf{w}^t + (1 - \nu) \mathbf{r}$,
if $\gamma > 0$, there is no advice update.

Advice Updates When $\sigma_t^h = 1$

- There are m advice vectors, $\mathbf{u}^{i,\star}$, $i = 1, \dots, m$
- At round t , the algorithm receives (\mathbf{x}^t, y_t)
- The **current hypothesis** is \mathbf{w}^t

same formulation as before; except need to take misclassification by \mathbf{w}^t into account since $\sigma_t^h = 1$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w} - \mathbf{w}^t\|_2^2 - \lambda y_t \mathbf{w}' \mathbf{x}^t + \frac{\mu}{2} \sum_{i=1}^m \|\boldsymbol{\eta}^i\|_2^2, \\ \text{s. t.} \quad & D_i' \mathbf{u}^{i,\star} + z_i \mathbf{w} + \boldsymbol{\eta}^i = 0, \quad i = 1, \dots, m. \end{aligned}$$

Advice Updates When $\sigma_t^h = 1$

Given advice-estimate \mathbf{r}^* , the **update** when the **current hypothesis does *not*** make a mistake is

$$\mathbf{w}^{t+1} = \nu(\mathbf{w}^t + \lambda y_t \mathbf{x}^t) + (1 - \nu) \mathbf{r}^*$$

Advice Updates When $\sigma_t^h = 1$

Given advice-estimate r^* , the update when the current hypothesis does **not** make a mistake is

$$\mathbf{w}^{t+1} = \nu(\mathbf{w}^t + \lambda y_t \mathbf{x}^t) + (1 - \nu) \mathbf{r}^*$$

same update rule as before;
contains an **additional term** that
updates according to the extent of
misclassification of \mathbf{x}^t by \mathbf{w}^t

A Unified Update Rule

Recall that misclassification by \mathbf{w}^t is indicated by

$$\sigma_t^h = \begin{cases} 1, & \text{if } y_t \mathbf{w}^{t'} \mathbf{x}^t \leq 0, \quad (\text{misclassification}) \\ 0, & \text{if } y_t \mathbf{w}^{t'} \mathbf{x}^t > 0, \quad (\text{correct classification}) \end{cases}$$

Then, updates are computed according to:

- when $\sigma_t^h = 0$, $\mathbf{w}^{t+1} = \nu \mathbf{w}^t + (1 - \nu) \mathbf{r}^*$
- when $\sigma_t^h = 1$, $\mathbf{w}^{t+1} = \nu(\mathbf{w}^t + \lambda y_t \mathbf{x}^t) + (1 - \nu) \mathbf{r}^*$

A Unified Update Rule

Recall that misclassification by \mathbf{w}^t is indicated by

$$\sigma_t^h = \begin{cases} 1, & \text{if } y_t \mathbf{w}^{t'} \mathbf{x}^t \leq 0, \quad (\text{misclassification}) \\ 0, & \text{if } y_t \mathbf{w}^{t'} \mathbf{x}^t > 0, \quad (\text{correct classification}) \end{cases}$$

Then, updates are computed according to:

- when $\sigma_t^h = 0$, $\mathbf{w}^{t+1} = \nu \mathbf{w}^t + (1 - \nu) \mathbf{r}^*$
- when $\sigma_t^h = 1$, $\mathbf{w}^{t+1} = \nu(\mathbf{w}^t + \lambda y_t \mathbf{x}^t) + (1 - \nu) \mathbf{r}^*$

This can be compactly written as

$$\mathbf{w}^{t+1} = \nu(\mathbf{w}^t + \sigma_t^h \lambda y_t \mathbf{x}^t) + (1 - \nu) \mathbf{r}^*$$

When Should Advice Updates Be Applied?

Recall that the *margin* of the updated hypothesis is

$$\gamma = \nu y_t \mathbf{w}^{t'} \mathbf{x}^t + (1 - \nu) y_t \mathbf{r}^{*'} \mathbf{x}^t$$

When Should Advice Updates Be Applied?

Recall that the **margin** of the updated hypothesis is

$$\gamma = \nu y_t \mathbf{w}^{t'} \mathbf{x}^t + (1 - \nu) y_t \mathbf{r}^{*'} \mathbf{x}^t$$

Now define, analogous to

$$\sigma_t^a = \begin{cases} 1, & \text{if } y_t(\nu \mathbf{w}^t + (1 - \nu) \mathbf{r})' \mathbf{x}^t \leq 0, \quad (\text{perform advice update}) \\ 0, & \text{if } y_t(\nu \mathbf{w}^t + (1 - \nu) \mathbf{r})' \mathbf{x}^t > 0. \quad (\text{no advice update}) \end{cases}$$

When Should Advice Updates Be Applied?

Recall that the **margin** of the updated hypothesis is

$$\gamma = \nu y_t \mathbf{w}^{t'} \mathbf{x}^t + (1 - \nu) y_t \mathbf{r}^{*'} \mathbf{x}^t$$

Now define, analogous to σ_t^h

$$\sigma_t^a = \begin{cases} 1, & \text{if } y_t(\nu \mathbf{w}^t + (1 - \nu) \mathbf{r})' \mathbf{x}^t \leq 0, \quad (\text{perform advice update}) \\ 0, & \text{if } y_t(\nu \mathbf{w}^t + (1 - \nu) \mathbf{r})' \mathbf{x}^t > 0. \quad (\text{no advice update}) \end{cases}$$

Now, we determine if there is an advice update:

- when $\sigma_t^a = 0$, $\mathbf{w}^{t+1} = \mathbf{w}^t + \lambda \sigma_t^h y_t \mathbf{x}^t$
- when $\sigma_t^a = 1$, $\mathbf{w}^{t+1} = \nu (\mathbf{w}^t + \lambda \sigma_t^h y_t \mathbf{x}^t) + (1 - \nu) \mathbf{r}^*$

The Adviceptron

- 1: **input** $(\mathbf{x}^t, y_t)_{t=1}^T$, advice sets $(D_i, \mathbf{d}^i, z_i)_{i=1}^m$, $\lambda, \mu > 0$
- 2: **pre-process** $(\mathbf{u}^{i,*}, \mathbf{w}^*)$ as optimal solution to advice KBSVM
- 3: let $\mathbf{r}^i = -z_i D_i' \mathbf{u}^{i,*}$, $\mathbf{r} = 1/m \sum_{i=1}^m \mathbf{r}^i$
- 4: let $\nu = 1/(1 + m\mu)$
- 5: let initial hypothesis, $\mathbf{w}^1 = \mathbf{w}^*$
- 6: **for** (\mathbf{x}^t, y_t) **do**
- 7: predict label $\hat{y}_t = \text{sign}(\mathbf{w}^{t'} \mathbf{x}^t)$
- 8: receive correct label y_t
- 9: compute σ_t^h and σ_t^a
- 10: **if** $\sigma_t^a = 0$ (**there is no advice update**) **then**
- 11: **update** $\mathbf{w}^{t+1} = \mathbf{w}^t + \lambda \sigma_t^h y_t \mathbf{x}^t$
- 12: **else if** $\sigma_t^a = 1$ (**there is an advice update**) **then**
- 13: **update** $\mathbf{w}^{t+1} = \nu(\mathbf{w}^t + \lambda \sigma_t^h y_t \mathbf{x}^t) + (1 - \nu)\mathbf{r}$
- 14: **end if**
- 15: **end for**

Outline

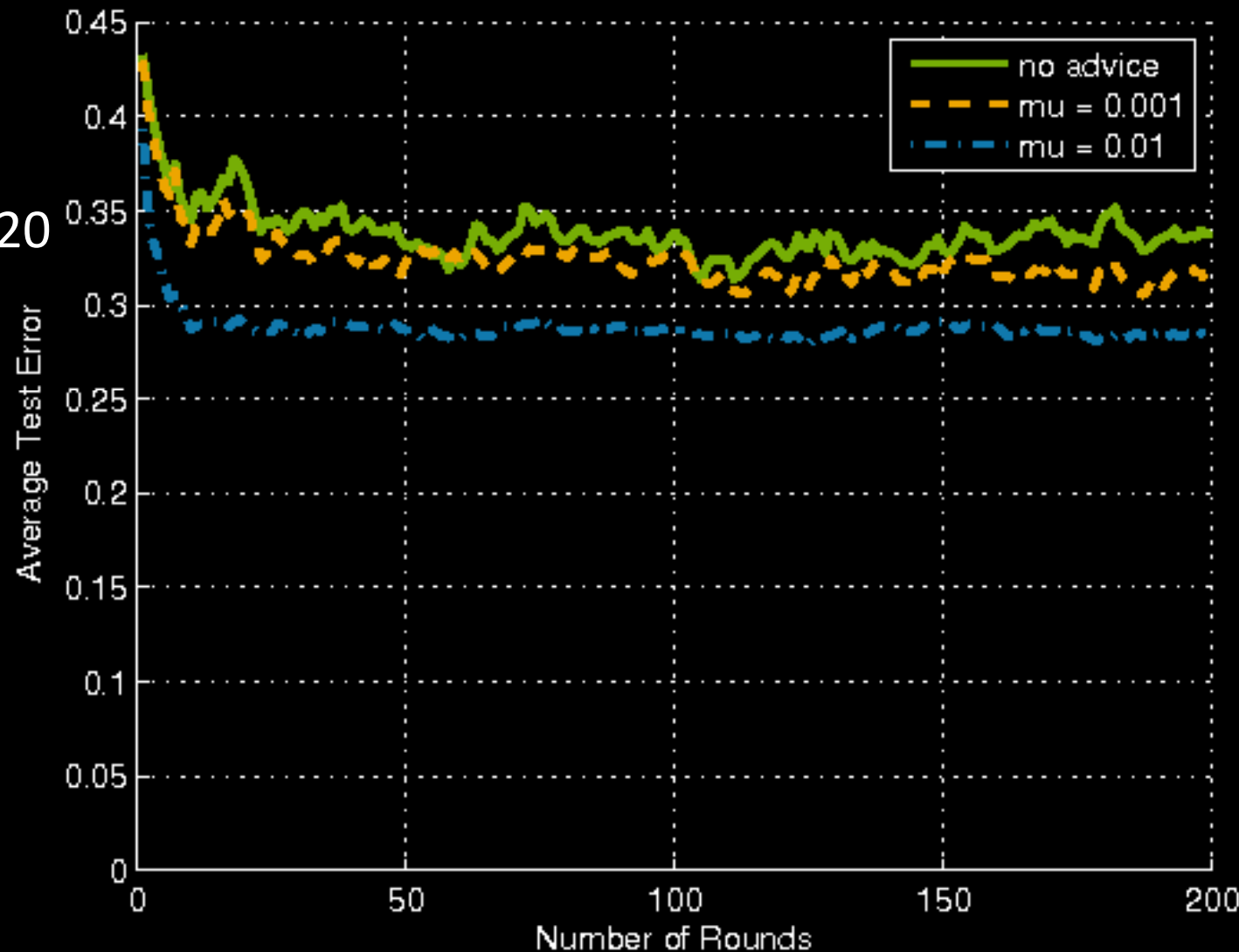
- Knowledge-Based Support Vector Machines
- The Adviceptron: Online KBSVMs
- ***A Real-World Task: Diabetes Diagnosis***
- A Real-World Task: Digit Recognition
- Conclusions

Diagnosing Diabetes

- Standard data set from UCI repository (768 x 8)
 - all patients at least 21 years old of Pima Indian heritage
 - features include **body mass index**, **blood glucose level**
- **Expert advice** for diagnosing diabetes from **NIH website on risks for Type-2 diabetes**
 - a person who is **obese** (characterized by BMI > 30) and has a **high blood glucose level** (> 126) is at a **strong risk for diabetes**
 $(\text{BMI} \geq 30) \wedge (\text{bloodglucose} \geq 126) \Rightarrow \text{diabetes}$
 - a person who is at **normal weight** (BMI < 25) and has **low blood glucose level** (< 100) is at a **low risk for diabetes**
 $(\text{BMI} \leq 25) \wedge (\text{bloodglucose} \leq 100) \Rightarrow \neg \text{diabetes}$

Diagnosing Diabetes: Results

- 200 examples for training, remaining for testing
- Results averaged over 20 randomized iterations

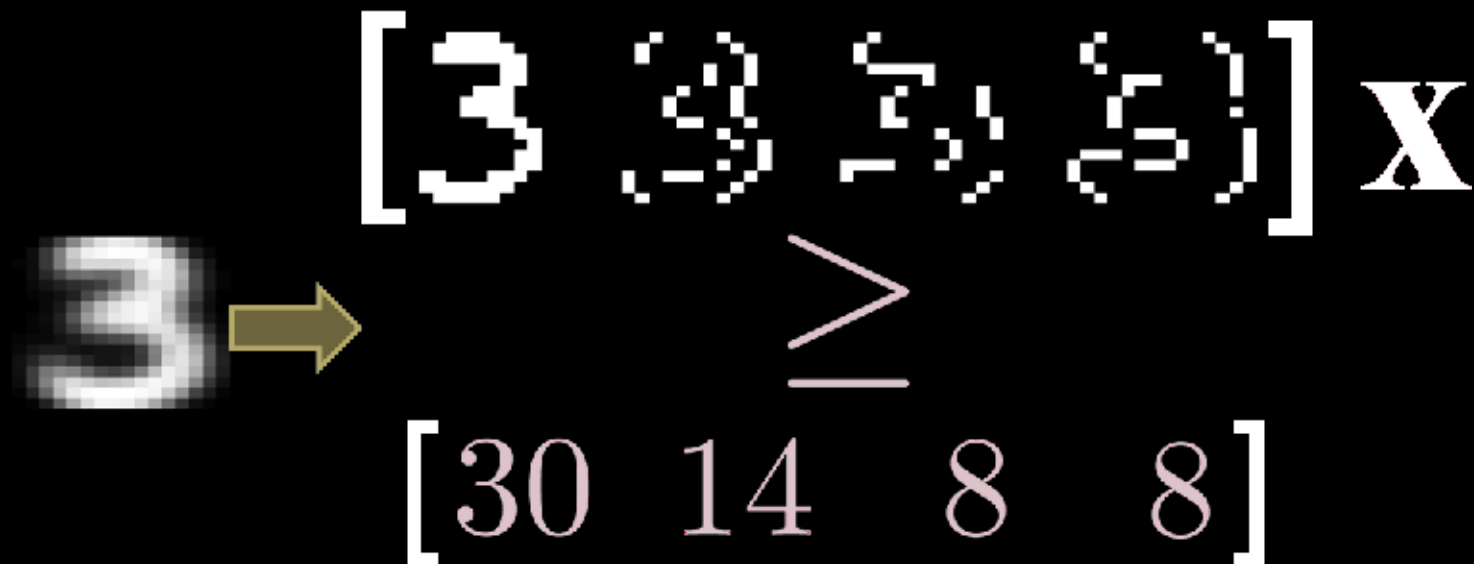


Outline

- Knowledge-Based Support Vector Machines
- The Adviceptron: Online KBSVMs
- A Real-World Task: Diabetes Diagnosis
- ***A Real-World Task: Digit Recognition***
- Conclusions

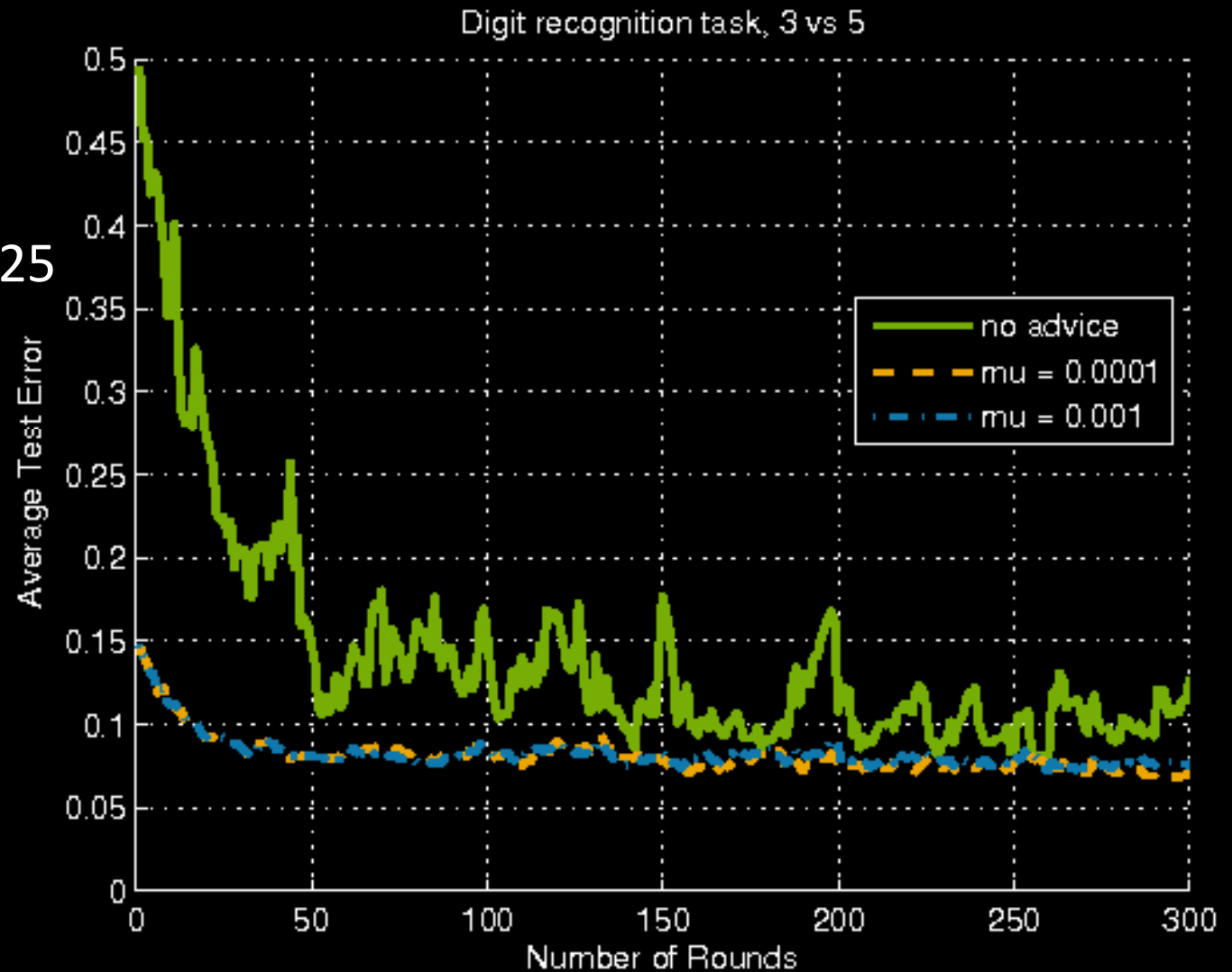
Digit Recognition

- Standard USPS data set
 - 9298 points, 16 x 16 greyscale images representing handwritten digits
- **Expert advice** for digit recognition based on “**canonical digits**” from 10 experts
 - **First 4 principal components along with threshold values**



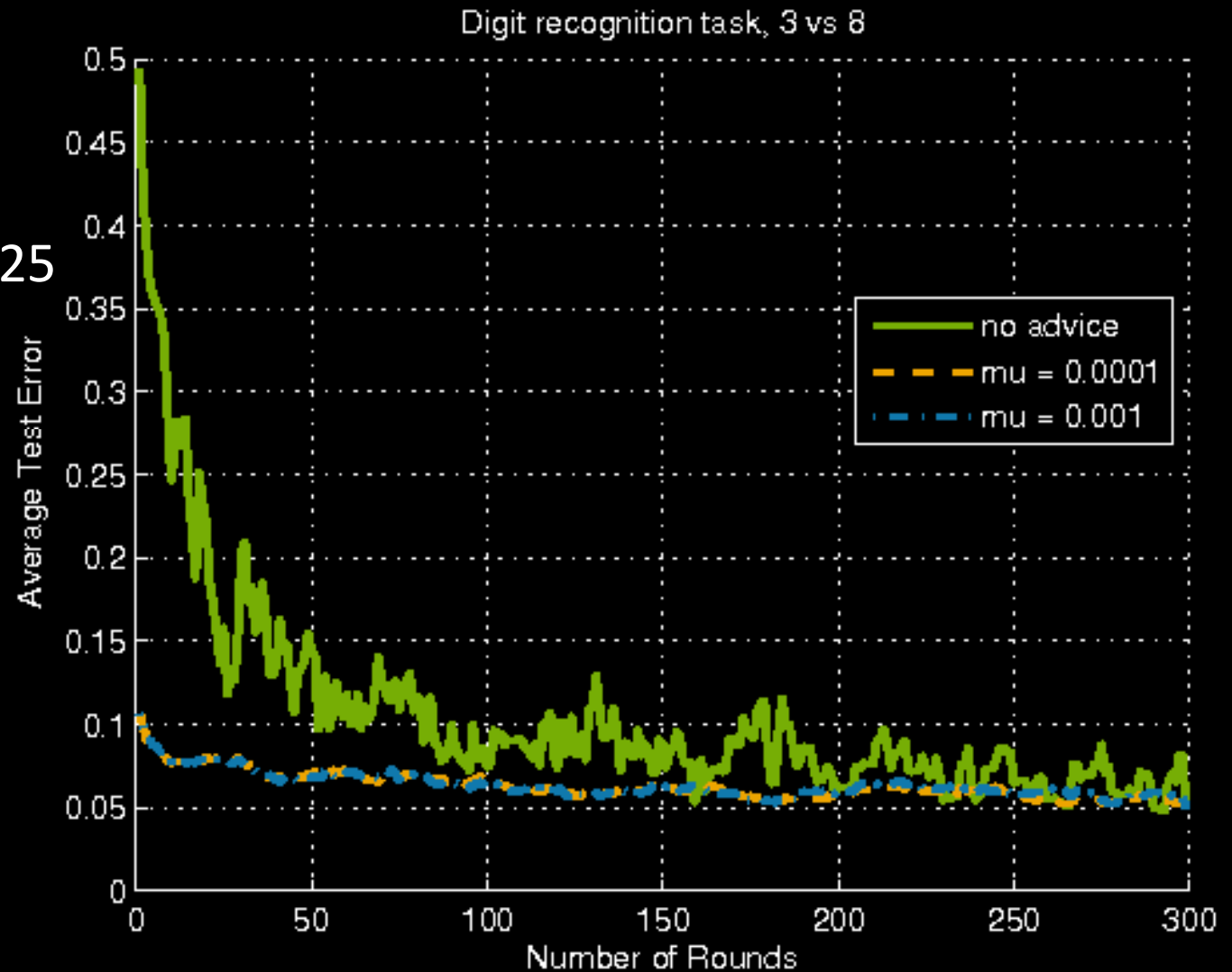
Digit Recognition: Results (3 vs 5)

- 500 examples for training, remaining for testing
- Results averaged over 25 randomized iterations



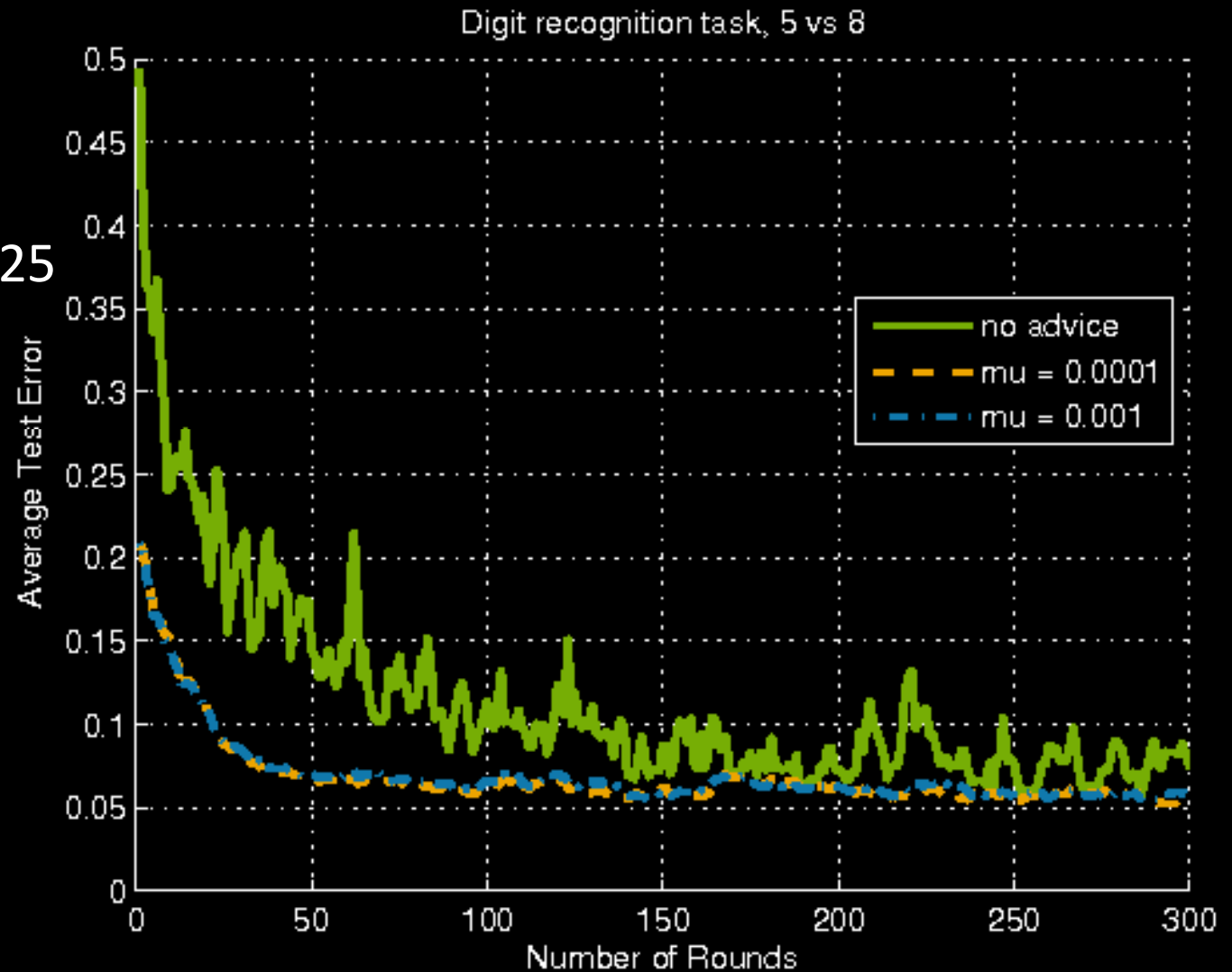
Digit Recognition: Results (3 vs 8)

- 500 examples for training, remaining for testing
- Results averaged over 25 randomized iterations



Digit Recognition: Results (5 vs 8)

- 500 examples for training, remaining for testing
- Results averaged over 25 randomized iterations



Analyzing the Adviceptron Empirically

- New online learning algorithm: ***the adviceptron***
- Makes use of prior knowledge in the form of (possibly imperfect) ***polyhedral advice***
- Performs ***simple, closed-form updates*** via passive-aggressive framework; scalable
- Good advice can help converge to a ***better solution with fewer examples***
- Further experiments in progress

Analyzing the Adviceptron Analytically

Let $S = \{(\mathbf{x}^t, y_t)\}_{t=1}^T$ be a sequence of examples with $(\mathbf{x}^t, y_t) \in \mathbb{R}^n \times \{\pm 1\}$.

Assume $\|\mathbf{x}^t\|_2 \leq X \quad \forall t$.

Let $A = \{(D_i, \mathbf{d}^i, z_i)\}_{i=1}^m$ be m advice sets with advice vectors $\mathbf{u}^{i,*} \geq 0$.

Denote

$\mathbf{r}^i = -z_i D_i' \mathbf{u}^{i,*}$ the i -th advice-estimate of the hypothesis

$\mathbf{r} = \frac{1}{m} \sum_{i=1}^m \mathbf{r}^i$, be the average advice-estimate.

Assume \mathbf{r} makes M_a mistakes on S .

If some $\mathbf{w}^* \in \mathbb{R}^n$ with $\|\mathbf{w}^* - \mathbf{r}\|_2 \leq R$ has a margin γ on S , the *Adviceptron* makes at most

$$M \leq \frac{X^2 R^2}{\gamma^2} (1 + (1 - \nu) M_a)$$

mistakes on S , where $\nu = 1/(1 + m\mu)$, $\mu > 0$.

Analyzing the Adviceptron Analytically

If some $\mathbf{w}^* \in \mathbb{R}^n$ with $\|\mathbf{w}^* - \mathbf{r}\|_2 \leq R$ has a margin γ on S , the *Adviceptron* makes at most

$$M \leq \frac{X^2 R^2}{\gamma^2} (1 + (1 - \nu) M_a)$$

mistakes on S , where $\nu = 1/(1 + m\mu)$, $\mu > 0$.

- **Smaller values of R tighten the bound** because \mathbf{w}^* is more “consistent” with the average advice-estimate \mathbf{r} . If $\mathbf{w}^* = \mathbf{r}$, we recover the original perceptron bound.
- **Fewer advice updates, M_a , tighten the bound.** More advice-consistent \mathbf{w}^t ensure that it is less likely there will be an advice update. This is because an advice update occurs **only** when

$$\nu y_t \mathbf{w}^{t'} \mathbf{x}^t + (1 - \nu) y_t \mathbf{r}' \mathbf{x}^t \leq 0.$$

If at the t -th trial, if \mathbf{w}^t is sufficiently influenced by the advice, there will be no mistake according to the advice ($\sigma_t^a = 0$) and no advice update.

References.

(Fung et al, 2003) G. Fung, O. L. Mangasarian, and J. W. Shavlik. *Knowledge-based support vector classifiers*. In S. Becker, S. Thrun & K. Obermayer, eds, NIPS, 15, pp. 521–528, 2003

(Crammer et al, 2006) K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. *Online passive-aggressive algorithms*. J. of Mach. Learn. Res., 7:551–585, 2006.

(Freund and Schapire, 1999) Y. Freund and R. E. Schapire. *Large margin classification using the perceptron algorithm*. Mach. Learn., 37(3):277–296, 1999.

Acknowledgements.

*We gratefully acknowledge support of DARPA under grant **HR0011-07-C-0060** and the NIH under grant **1-R01-LM009731-01**.*

Views and conclusions contained in this document are those of the authors and do not necessarily represent the official opinion or policies, either expressed or implied of the US government or of DARPA.

KBSVMs: Deriving The Advice Constraints

We assume an expert provides **polyhedral advice** of the form

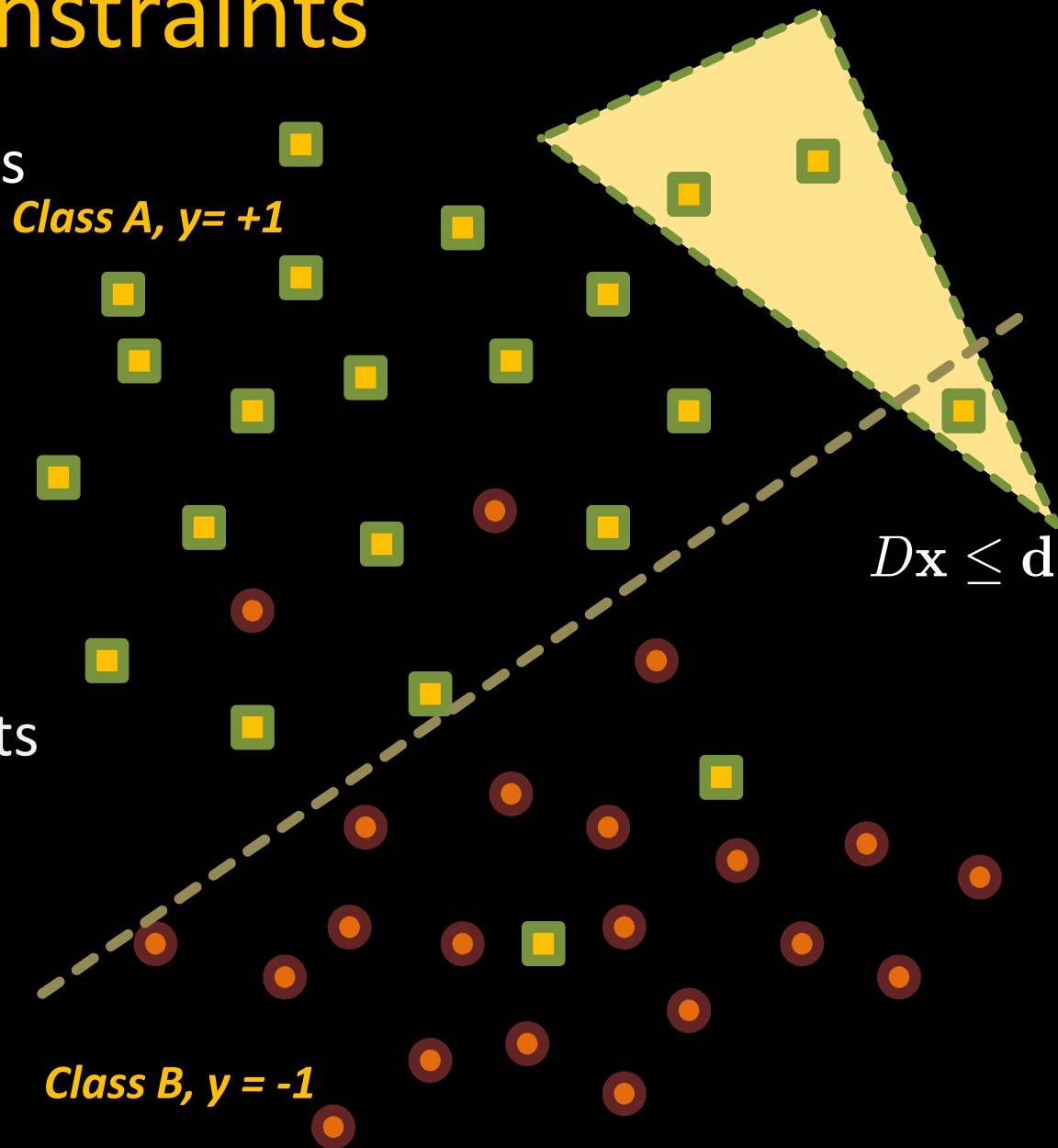
$$D\mathbf{x} \leq \mathbf{d} \Rightarrow \mathbf{w}'\mathbf{x} \geq b$$

We know $p \Rightarrow q$ is equivalent to $\neg p \vee q$

If $\neg p \vee q$ has a solution then its **negation has no solution** or,

$$\begin{aligned} D\mathbf{x} - \mathbf{d}\tau &\leq 0, \\ \mathbf{w}'\mathbf{x} - b\tau &< 0, \\ -\tau &< 0 \end{aligned}$$

has no solution (\mathbf{x}, τ) .



KBSVMs: Deriving The Advice Constraints

If the following system

$$\begin{aligned} D\mathbf{x} - \mathbf{d}\tau &\leq 0, \\ \mathbf{w}'\mathbf{x} - b\tau &< 0, \\ -\tau &< 0 \end{aligned}$$

has no solution (\mathbf{x}, τ) , then by

Motzkin's Theorem of the Alternative, the following system

$$\begin{aligned} D'\mathbf{u} + \mathbf{w} &= 0, \\ -\mathbf{d}'\mathbf{u} - b &\geq 0, \\ \mathbf{u} &\geq 0 \end{aligned}$$

has a solution \mathbf{u} .

