Asymmetric Support Vector Machines

Low False-Positive Learning Under the User Tolerance

Brandon, Shan-Hung Wu, Keng-Pei Lin, Chung-Min Chen, and Ming-Syan Chen

Telcordia Technologies

2010/05/21

Introduction (1/3)

- Who is Telcordia?
  - Formerly Bellcore
  - Established Applied Research Center in Taiwan 2004
  - Focuses on Telematics R&D

Introduction (1/3)

- Who is Telcordia?
  - Formerly Bellcore
  - Established Applied Research Center in Taiwan 2004
  - Focuses on Telematics R&D
- Who am I?
  - Technical Lead of the **Data Management & Situation Awareness** group
  - Detect/predict dangerous situations for drivers/administrators

Introduction (1/3)

- Who is Telcordia?
  - Formerly Bellcore
  - Established Applied Research Center in Taiwan 2004
  - Focuses on Telematics R&D
- Who am I?
  - Technical Lead of the **Data Management & Situation Awareness** group
  - Detect/predict dangerous situations for drivers/administrators
- Why am I standing here?
  - We use SVM
  - A classifier is required to produce a **very low False-Positive (FP) rate**
  - Actually, many real-world applications are particularly sensitive to the wrong predictions of a certain class
    - E.g., Spam filtering, facial image recognition, network intrusion detection, computer-aided disease diagnosis, etc.

Introduction (2/3)

- For the Support Vector Machine (SVM) classifier, two common techniques are applied prior and posterior to the main training process respectively **without modifying the algorithm**

Introduction (2/3)

- For the Support Vector Machine (SVM) classifier, two common techniques are applied prior and posterior to the main training process respectively **without modifying the algorithm**
  1. **Parameter tuning**: require prior knowledge or long training process

Introduction (2/3)

- For the Support Vector Machine (SVM) classifier, two common techniques are applied prior and posterior to the main training process respectively **without modifying the algorithm**
  1. **Parameter tuning**: require prior knowledge or long training process
  2. **Thresholding**: suffer from trade-off between maximizing TPs and minimizing FPs

Introduction (3/3)

- In this paper, we propose Asymmetric Support Machine (ASVM)
  - Take into account the false-positive rate and user tolerance **natively** in the objective formulation

Introduction (3/3)

- In this paper, we propose Asymmetric Support Machine (ASVM)
  - Take into account the false-positive rate and user tolerance **natively** in the objective formulation
- Give more insight into a dataset
  - The values of ASVM's parameters can reflect the portion of outliers from each of the classes

Introduction (3/3)

- In this paper, we propose Asymmetric Support Machine (ASVM)
  - Take into account the false-positive rate and user tolerance **natively** in the objective formulation
- Give more insight into a dataset
  - The values of ASVM's parameters can reflect the portion of outliers from each of the classes
- Give either
  - 6.4% improvement in AUC when compared to the Thresholding, or
  - An order faster training time when compared to the Parameter Tuning

Agenda

1. Preliminaries
   - The SVM Classifier
   - Current Ways to Reduce the FP Rate
2. Asymmetric Support Vector Machine (ASVM)
   - Objective Formulation & Rationale
   - A Toy Example
3. Effects of Parameters
4. Performance Evaluation
5. Conclusions

Agenda

1. Preliminaries
   - The SVM Classifier
   - Current Ways to Reduce the FP Rate
2. Asymmetric Support Vector Machine (ASVM)
   - Objective Formulation & Rationale
   - A Toy Example
3. Effects of Parameters
4. Performance Evaluation
5. Conclusions

The SVM Classifier: Basic Concept (1/3)

- Target: to associate each testing instance with a real value
  - Those with values > 0 will be predicted as positive
  - Those with values < 0 will be predicted as negative

The SVM Classifier: Basic Concept (1/3)

- Target: to associate each testing instance with a real value
  - Those with values > 0 will be predicted as positive
  - Those with values < 0 will be predicted as negative
- How? Given m training instances $(\mathbf{x}_i, y_i)$, $i = 1 \cdots m$, where $\mathbf{x}_i$ are vectors of features and $y_i \in \{\pm 1\}$ are labels,
- Find a linear function $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$ such that

$$\begin{array}{ll} \langle \mathbf{w}, \mathbf{x}_i \rangle + b > 0, & \text{if } y_i = 1, \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b < 0, & \text{otherwise,} \end{array}$$

$\forall i = 1 \cdots m.$

The SVM Classifier: Basic Concept (1/3)

- Target: to associate each testing instance with a real value
  - Those with values > 0 will be predicted as positive
  - Those with values < 0 will be predicted as negative
- How? Given m training instances $(\mathbf{x}_i, y_i)$, $i = 1 \cdots m$, where $\mathbf{x}_i$ are vectors of features and $y_i \in \{\pm 1\}$ are labels,
- Find a linear function $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$ such that

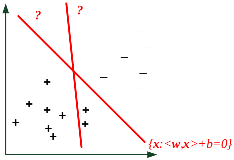
$$\begin{array}{ll} \langle \mathbf{w}, \mathbf{x}_i \rangle + b > 0, & \text{if } y_i = 1, \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b < 0, & \text{otherwise,} \end{array}$$

$\forall i = 1 \cdots m$.

- Once $\mathbf{w}$ and b are determined, simply use $\operatorname{sgn}(\langle \mathbf{w}, \widetilde{\mathbf{x}} \rangle + b)$ to predict the label of a testing instance $\widetilde{\mathbf{x}}$

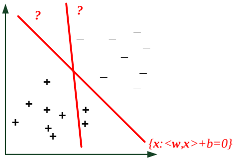
The SVM Classifier: Basic Concept (2/3)

- Aren't there many possible alternatives for **w** and b? Yes!

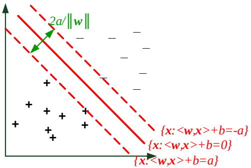


The SVM Classifier: Basic Concept (2/3)

- Aren't there many possible alternatives for **w** and *b*? Yes!



- SVM classifier chooses the ones **that result in the largest margin**

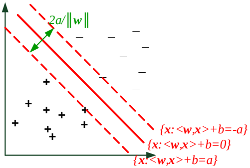


The SVM Classifier: Basic Concept (3/3)

- To describe a margin, we let

$$\begin{array}{ll} \langle \mathbf{w}, \mathbf{x}_i \rangle + b > a, & \text{if } y_i = 1, \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b < -a, & \text{otherwise,} \end{array}$$

  for each training instance

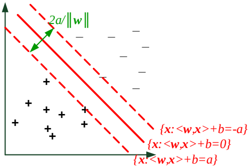


The SVM Classifier: Basic Concept (3/3)

- To describe a margin, we let

$$\begin{aligned} \langle \mathbf{w}, \mathbf{x}_i \rangle + b > a, & \quad \text{if } y_i = 1, \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b < -a, & \quad \text{otherwise,} \end{aligned}$$

  for each training instance



- Observe that the margin is proportional to the inverse of $\|\mathbf{w}\| = \langle \mathbf{w}, \mathbf{w} \rangle^{1/2}$, we form the following objective:

$$\arg\min_{\mathbf{w}, b} \langle \mathbf{w}, \mathbf{w} \rangle \,,$$

$$\text{subject to } y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, \, \forall i = 1 \cdots m.$$

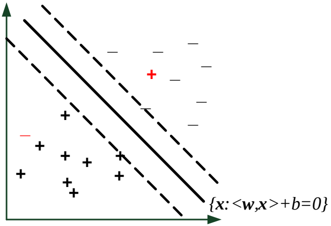
  - Note the constant a can be any value, here we choose $a = 1$ for simplicity

The SVM Classifier: Coping with Overlapped Data (1/2)

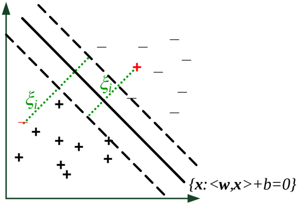
- The above objective can be transformed into a quadratic optimization problem, which can be solved in polynomial time

The SVM Classifier: Coping with Overlapped Data (1/2)

- The above objective can be transformed into a quadratic optimization problem, which can be solved in polynomial time
- Everything looks fine, but how about this?

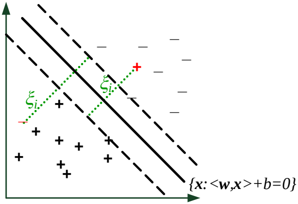


- Solution: to allow **slacks** that fall outside the regions they ought to be
  - We let $y_i \langle \mathbf{w}, \mathbf{x}_i \rangle + b > 1 - \xi_i$ for each training instance, where $\xi_i \geq 0$
  - Slacks (i.e., those with positive ξ_i) are usually treated as noise or outliers



The SVM Classifier: Coping with Overlapped Data (2/2)

- Solution: to allow **slacks** that fall outside the regions they ought to be
  - We let $y_i \langle \mathbf{w}, \mathbf{x}_i \rangle + b > 1 - \xi_i$ for each training instance, where $\xi_i \geq 0$
  - Slacks (i.e., those with positive ξ_i) are usually treated as noise or outliers



- SVM classifier favors a larger margin **but also fewer slacks**:

$$\arg\min_{\mathbf{w},b,\xi} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{m} \xi_i,$$

$$\text{subject to } y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \text{ and } \xi_i \geq 0.$$

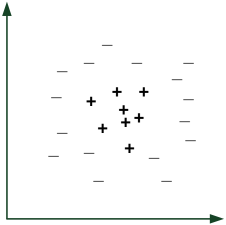
  - The parameter C controls the trade-off between maximizing the margin and minimizing the costs of slacks

The SVM Classifier: Kernel Trick (1/3)

- The above objective can still be solved in polynomial time

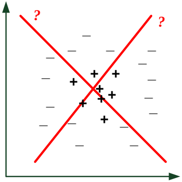
The SVM Classifier: Kernel Trick (1/3)

- The above objective can still be solved in polynomial time
- Now, how about this?



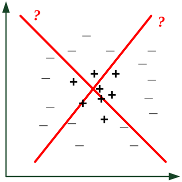
The SVM Classifier: Kernel Trick (2/3)

- The solutions at right are far from ideal, even the slacks are allowed
  - We know the answer is a "circle," but it is **not linear** anymore

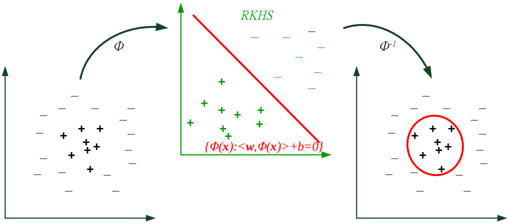


The SVM Classifier: Kernel Trick (2/3)

- The solutions at right are far from ideal, even the slacks are allowed
  - We know the answer is a "circle," but it is **not linear** anymore



- Do we have to give up all the concept just learned? Nope, if we do some trick upon the data:



The SVM Classifier: Kernel Trick (3/3)

- SVM classifier can operate in a high-dimensional Reproducing Kernel Hilbert Space (RKHS):

$$\arg\min_{\mathbf{w},b,\xi} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{m} \xi_i,$$
$$\text{subject to } y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, \tag{1}$$

$\forall i = 1 \cdots m$.

The SVM Classifier: Kernel Trick (3/3)

- SVM classifier can operate in a high-dimensional Reproducing Kernel Hilbert Space (RKHS):

$$\arg\min_{\mathbf{w},b,\xi} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{m} \xi_i,$$

$$\text{subject to } y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, \qquad (1)$$

$\forall i = 1 \cdots m$.

  - A testing instance $\tilde{\mathbf{x}}$ is predicted as positive iff $\langle \mathbf{w}, \Phi(\tilde{\mathbf{x}}_i) \rangle + b > 0$

The SVM Classifier: Kernel Trick (3/3)

- SVM classifier can operate in a high-dimensional Reproducing Kernel Hilbert Space (RKHS):

$$\arg\min_{\mathbf{w},b,\xi} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{m} \xi_i,$$

$$\text{subject to } y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, \tag{1}$$

$\forall i = 1 \cdots m.$

  - A testing instance $\widetilde{\mathbf{x}}$ is predicted as positive iff $\langle \mathbf{w}, \Phi(\widetilde{\mathbf{x}}_i) \rangle + b > 0$
- Moreover, the term $\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle$ can be efficiently calculated **within linear time to the original space** using a kernel function
  - This is known as the **kernel trick**

Current Ways to Reduce the FP Rate (1/2)

- Two common techniques are applied prior and posterior to the training process of SVM respectively
  - The main SVM algorithm is untouched

Current Ways to Reduce the FP Rate (1/2)

- Two common techniques are applied prior and posterior to the training process of SVM respectively
  - The main SVM algorithm is untouched
- **Thresholding** (posterior): a testing instance $\widetilde{\mathbf{x}}$ is predicted as positive iff $\langle \mathbf{w}, \Phi(\widetilde{\mathbf{x}}_i) \rangle + b > t$, where $t \geq 0$ is a threshold

Current Ways to Reduce the FP Rate (1/2)

- Two common techniques are applied prior and posterior to the training process of SVM respectively
  - The main SVM algorithm is untouched
- **Thresholding** (posterior): a testing instance $\widetilde{\mathbf{x}}$ is predicted as positive iff $\langle \mathbf{w}, \Phi(\widetilde{\mathbf{x}}_i) \rangle + b > t$, where $t \geq 0$ is a threshold
  - The larger the value of t, the less chance a false-positive occurs

Current Ways to Reduce the FP Rate (1/2)

- Two common techniques are applied prior and posterior to the training process of SVM respectively
  - The main SVM algorithm is untouched
- **Thresholding** (posterior): a testing instance $\widetilde{\mathbf{x}}$ is predicted as positive iff $\langle \mathbf{w}, \Phi(\widetilde{\mathbf{x}}_i) \rangle + b > t$, where $t \geq 0$ is a threshold
  - The larger the value of t, the less chance a false-positive occurs
  - However, fewer true-positives may be identified

Current Ways to Reduce the FP Rate (1/2)

- Two common techniques are applied prior and posterior to the training process of SVM respectively
  - The main SVM algorithm is untouched
- **Thresholding** (posterior): a testing instance $\widetilde{\mathbf{x}}$ is predicted as positive iff $\langle \mathbf{w}, \Phi(\widetilde{\mathbf{x}}_i) \rangle + b > t$, where $t \geq 0$ is a threshold
  - The larger the value of t, the less chance a false-positive occurs
  - However, fewer true-positives may be identified
- Such a technique suffers from an unwanted trade-off between minimizing the false-positive rate and maximizing the true-positive rate

Current Ways to Reduce the FP Rate (1/2)

- **Parameter Tuning** (anterior): differentiate the cost C of the slack variables in Eq. (1) (ref., $\arg\min_{\mathbf{w},b,\xi} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{m} \xi_i$)
  - Could be either

$$\arg\min_{\mathbf{w},b,\xi} \langle \mathbf{w}, \mathbf{w} \rangle + C^{+} \sum_{y_i=1} \xi_i + C^{-} \sum_{y_i=-1} \xi_i$$

or

$$\arg\min_{\mathbf{w},b,\xi} \langle \mathbf{w}, \mathbf{w} \rangle + \sum_{i=1}^{m} C_i \xi_i$$

Current Ways to Reduce the FP Rate (1/2)

- **Parameter Tuning** (anterior): differentiate the cost C of the slack variables in Eq. (1) (ref., $\arg\min_{\mathbf{w},b,\xi} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{m} \xi_i$)
  - Could be either

$$\arg\min_{\mathbf{w},b,\xi} \langle \mathbf{w}, \mathbf{w} \rangle + C^{+} \sum_{y_i=1} \xi_i + C^{-} \sum_{y_i=-1} \xi_i$$

or

$$\arg\min_{\mathbf{w},b,\xi} \langle \mathbf{w}, \mathbf{w} \rangle + \sum_{i=1}^{m} C_i \xi_i$$

- Such a technique require either domain-specific knowledge or time-consuming searches for the optimal combination of the costs

- **Parameter Tuning** (anterior): differentiate the cost C of the slack variables in Eq. (1) (ref., $\arg\min_{\mathbf{w},b,\xi} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{m} \xi_i$)
  - Could be either

$$\arg\min_{\mathbf{w},b,\xi} \langle \mathbf{w}, \mathbf{w} \rangle + C^{+} \sum_{y_i=1} \xi_i + C^{-} \sum_{y_i=-1} \xi_i$$

or

$$\arg\min_{\mathbf{w},b,\xi} \langle \mathbf{w}, \mathbf{w} \rangle + \sum_{i=1}^{m} C_i \xi_i$$

- Such a technique require either domain-specific knowledge or time-consuming searches for the optimal combination of the costs
- There is a basic need for a new SVM classifier that takes into account the FP rate

Agenda

Asymmetric Support Vector Machine

- In this paper, we propose Asymmetric Support Vector Machine (ASVM) that is able to model the false-positives in its objective

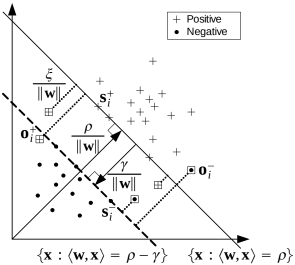
Asymmetric Support Vector Machine

- In this paper, we propose Asymmetric Support Vector Machine (ASVM) that is able to model the false-positives in its objective
- ASVM is asymmetric in the sense that it maximizes the margin between the negative class and the **core** (i.e., high confidence subset) of the positive class
  - Basically, the smaller the core (i.e., the higher the confidence), the less chance a false-positive may occur

Asymmetric Support Vector Machine

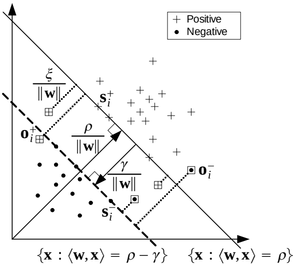
- In this paper, we propose Asymmetric Support Vector Machine (ASVM) that is able to model the false-positives in its objective
- ASVM is asymmetric in the sense that it maximizes the margin between the negative class and the **core** (i.e., high confidence subset) of the positive class
  - Basically, the smaller the core (i.e., the higher the confidence), the less chance a false-positive may occur
- How? We introduce a **core-margin** in addition to the traditional class-margin

Objective Formulation (1/2)



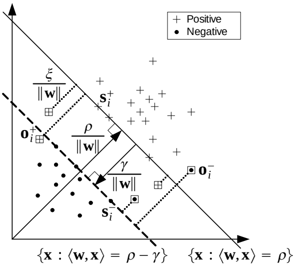
- The class-margin, $\gamma / \|\mathbf{w}\|$, is maximized to enhance the classification performance

Objective Formulation (1/2)



- The class-margin, $\gamma / \|\mathbf{w}\|$, is maximized to enhance the classification performance
- The core-margin, $\rho / \|\mathbf{w}\|$, is maximized (in RKHS) to capture the core of the positive class

Objective Formulation (1/2)



- The class-margin, $\gamma / \|\mathbf{w}\|$, is maximized to enhance the classification performance
- The core-margin, $\rho / \|\mathbf{w}\|$, is maximized (in RKHS) to capture the core of the positive class
- Since the class- and core-margins are maximized simultaneously, this approach avoids the trade-off between maximizing TP and minimizing FP in Thresholding

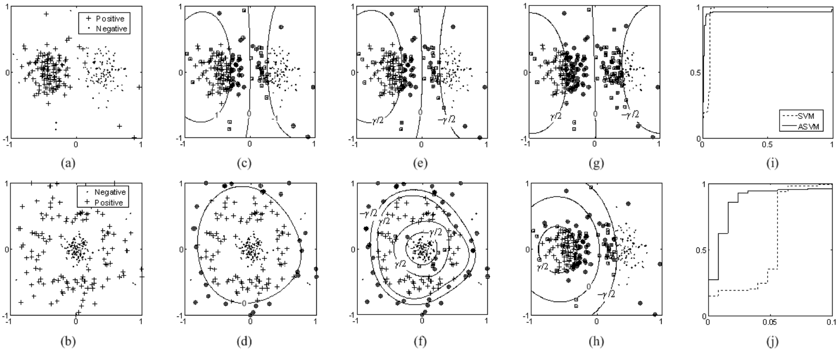
Objective Formulation (2/2)

- The asymmetric objective:

$$\arg\min_{\mathbf{w},\rho,\gamma,\xi} \langle \mathbf{w}, \mathbf{w} \rangle - \rho - \frac{\mu}{\tau}\gamma + \frac{1}{\tau m}\sum_{i=1}^{m} \xi_i, \tag{2}$$

subject to $y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle - \rho) + \frac{1}{2}(y_i - 1)\gamma \geq -\xi_i$, $\xi_i \geq 0$, and $\gamma \geq 0$.

  - The effects of parameters, μ and τ, will be explained later
- A testing instance $\widetilde{\mathbf{x}}$ is predicted as positive iff $\langle \mathbf{w}, \Phi(\widetilde{\mathbf{x}}_i) \rangle > \rho$



(a) (c) (e) (g) (i)

(b) (d) (f) (h) (j)

Agenda

Effects of Parameters (1/2)

- We study the effects of the ASVM parameters and observe their linkage to the empirical measure over the **portion of outliers**
- Let $\Pr^{emp}(\mathbf{o}_i^+)$ and $\Pr^{emp}(\mathbf{o}_i^-)$ denote the portion of outliers from the positive and negative classes respectively in the training instances

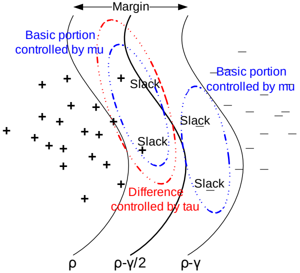
Theorem (Effect of τ)

The difference $\Pr^{emp}(\mathbf{o}_i^+) - \Pr^{emp}(\mathbf{o}_i^-)$ *converges almost surely to* τ, *i.e.,* $\Pr(\lim_{m\to\infty}(\Pr^{emp}(\mathbf{o}_i^+) - \Pr^{emp}(\mathbf{o}_i^-)) = \tau) = 1.$

Corollary (Effect of μ)

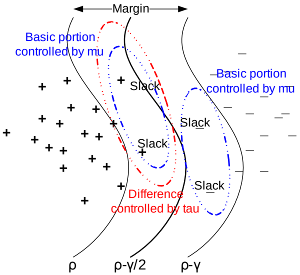
$\Pr^{emp}(\mathbf{o}_i^-)$ *converges almost surely to* μ.

Effects of Parameters (2/2)



- The above theorems give more insight into the datasets than traditional SVM does
  - Allow ASVM to incorporate with the prior knowledge

Effects of Parameters (2/2)



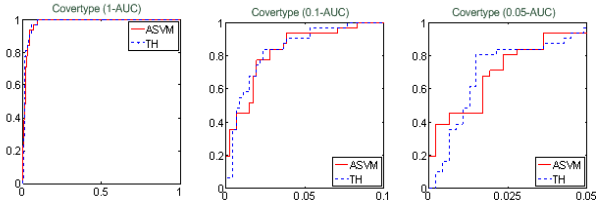
- The above theorems give more insight into the datasets than traditional SVM does
  - Allow ASVM to incorporate with the prior knowledge

- What really matters? We observe that the control over FP rate can be characterized by **a dedicated parameter**, τ, rather than distributed among C^+ and C^- (or C_i) as in Parameter Tuning
  - This enables significant reduction in training time, as we will see next

Agenda

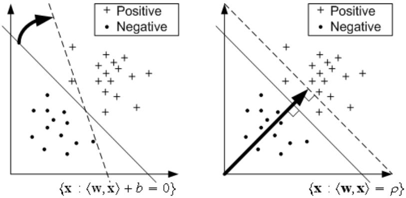
Performance Evaluation (1/2)

- As compared with Thresholding,
  - Generally, about 6% improvement in AUC
  - Stay as the best classifier in the low-FP region of the ROC curve



Performance Evaluation (2/2)

- As compared with Parameter Tuning
  - Render comparable performance results
  - Require only $O(m^2)$ training times in searching the best combination of parameters, an order faster than that ($O(m^3)$) of Parameter Tuning



Agenda

Conclusions

- We propose the Asymmetric Support Vector Machine
  - Capture the core of the positive class to increase the confidence of positive predictions

Conclusions

- We propose the Asymmetric Support Vector Machine
  - Capture the core of the positive class to increase the confidence of positive predictions
- The class- and core-margins are maximized at the same time that avoids the traditional trade-off
  - Give 6.4% improvement in AUC when compared to the Thresholding

Conclusions

- We propose the Asymmetric Support Vector Machine
  - Capture the core of the positive class to increase the confidence of positive predictions
- The class- and core-margins are maximized at the same time that avoids the traditional trade-off
  - Give 6.4% improvement in AUC when compared to the Thresholding
- The effect of asymmetry is described by a dedicated parameter, τ
  - Achieve an order faster training time when compared to the Parameter Tuning

Thanks

Q&A