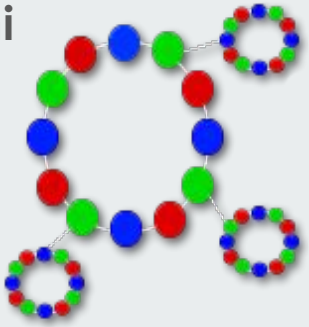
Learning Multiclassifiers with Predictive Features that Vary with Data Distribution

Omid Askarisichani
Xuan-Hong Dang
Ambuj Singh

13 December 2018

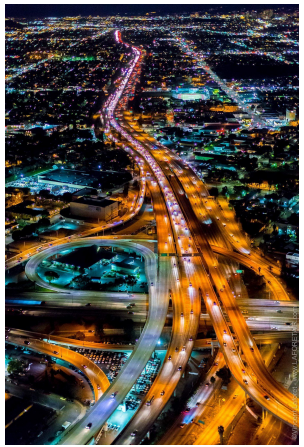


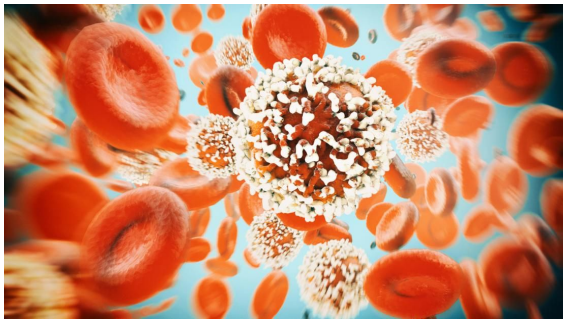
Outline

- Motivation
- Background
- Idea
- Datasets
- Proposed method
- Experimental results

Motivation: many big data are not-homogenous

- Data naturally have multiple known clusters with different distributions.
- We know the similarity among clusters.





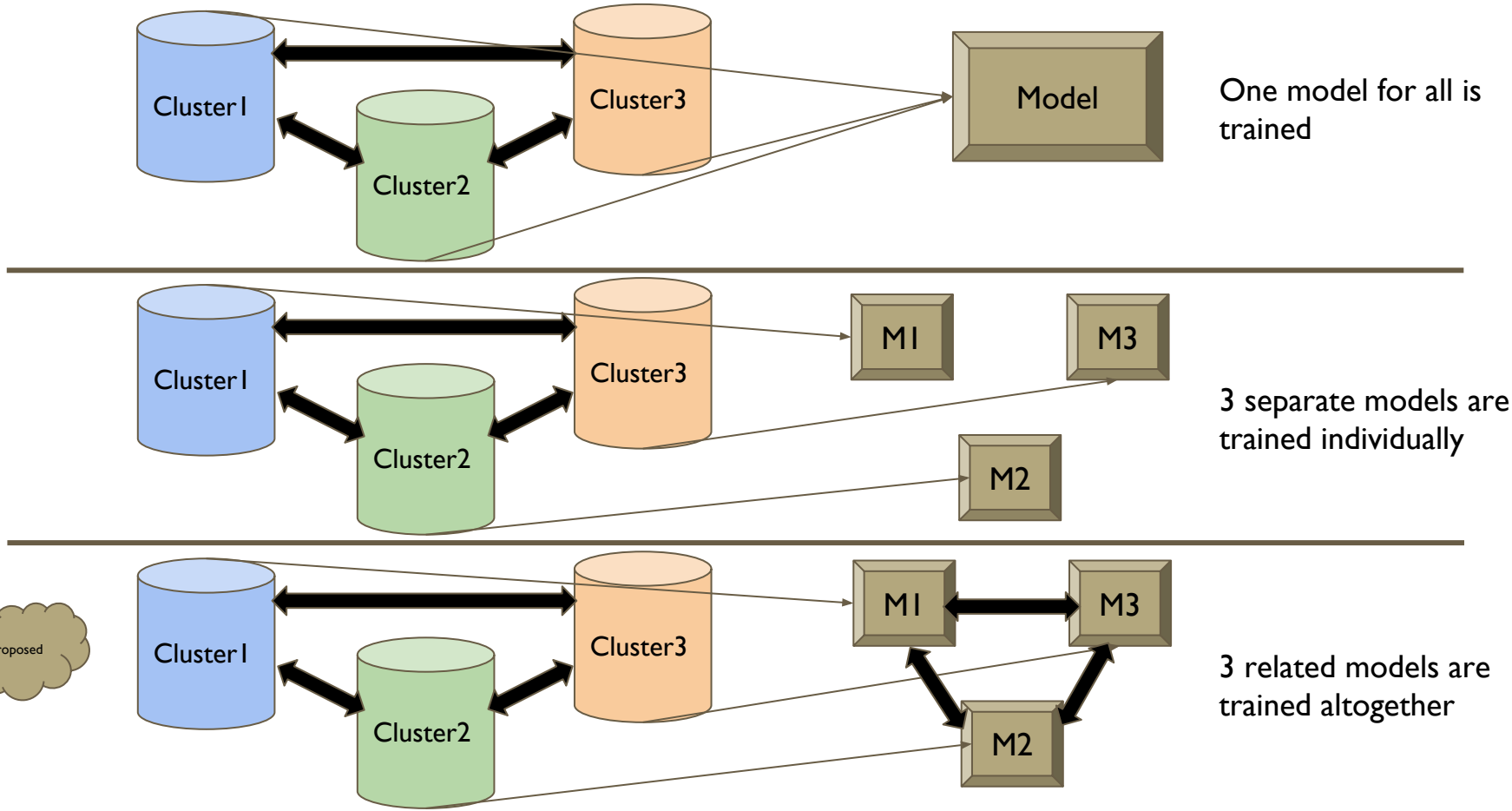
Examples:

- All temporal data, i.e. traffic network.
- Difficulty levels in a league/game.
- Progression of diseases.

General approaches:

- Model on entire data?
- Model or separate clusters?

Example


Cluster1
Cluster2
Cluster3
Model
One model for all is trained
Cluster1
Cluster2
Cluster3
M1
M2
M3
3 separate models are trained individually
Proposed
Cluster1
Cluster2
Cluster3
M1
M2
M3
3 related models are trained altogether

Motivation: prediction accuracy vs. model interpretability

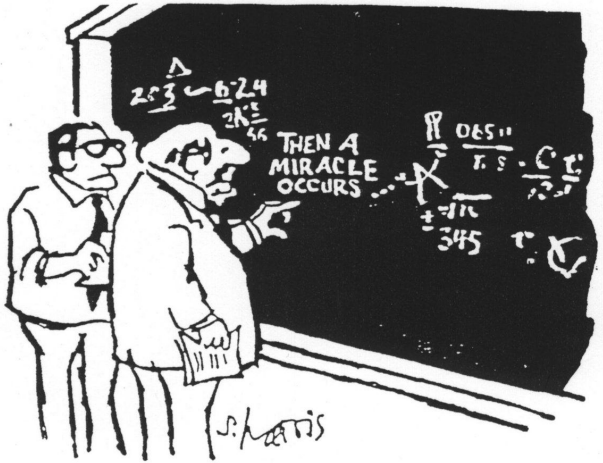
White box vs. black box trade-off:

Although predictive power remains important, a model that learns a succinct set of input features would be of high interest.

Model interpretability:

More convincing, more maintainable.

Q: Can we learn a model that takes advantage of similarities among clusters while finding an interpretable set of features that can vary over clusters and gives a high accuracy?



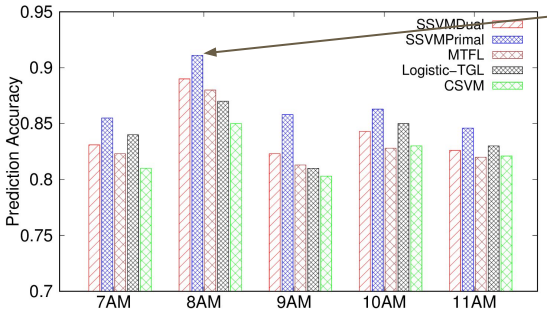
I think you should be a little more specific, here in Step 2

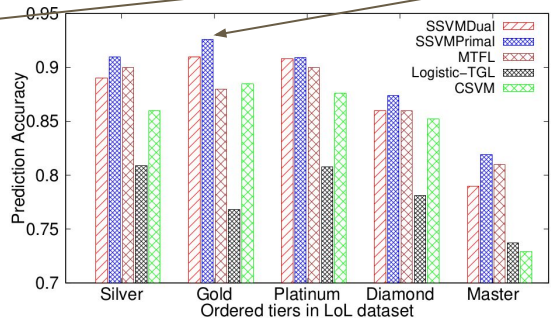
Early conclusion

The proposed model:

- Takes advantage of all similarities in the data
- Discovers a succinct set of features
- Finds different sets of features for every cluster
- Achieves not only high accuracy but also gives insights about data
- Is applied on two different domain real-world datasets
- In every cluster in every dataset surpasses the baseline methods

proposed





Outline

- Motivation
- Background
- Idea
- Datasets
- Proposed method
- Experimental results

Multi-task learning

Goal: improving the predictive performance of multiple related tasks by exploiting intrinsic relationships among them

Categories:

- Implicit: enforcing common structure (i.e. low rank subspaces) [1, 2]
- Explicit: enforcing similarity among parameters (i.e. similar to average) [3, 4, 5]

Issue:

- Similar sets of features for different clusters
- Simple formulation such as Linear or Logistic Regression
- Large set of features

[1] Lee, S., Zhu, J., Xing, E.P. "Adaptive multi-task lasso: with application to eqtl detection.", NIPS, (2010)
[2] Liu, J., Ji, S., Ye, J. "Multi-task feature learning via efficient l-2, 1-norm minimization.", 25th conference on uncertainty in artificial intelligence, (2009)
[3] Zhao, L., Q. Sun, J. Ye, F. Chen, C. Lu, and N. Ramakrishnan. "Multi-task learning for spatio-temporal event forecasting.", SIGKDD, (2015)
[4] Evgeniou, T., Pontil, M. "Regularized multi–task learning.", SIGKDD (2004)
[5] Zhou, J., L. Yuan, J. Liu, and J. Ye. "A multi-task learning formulation for predicting disease progression.", SIGKDD, (2011)

Outline

- Motivation
- Background
- Idea
- Datasets
- Proposed method
- Experimental results

SVM problem formulation

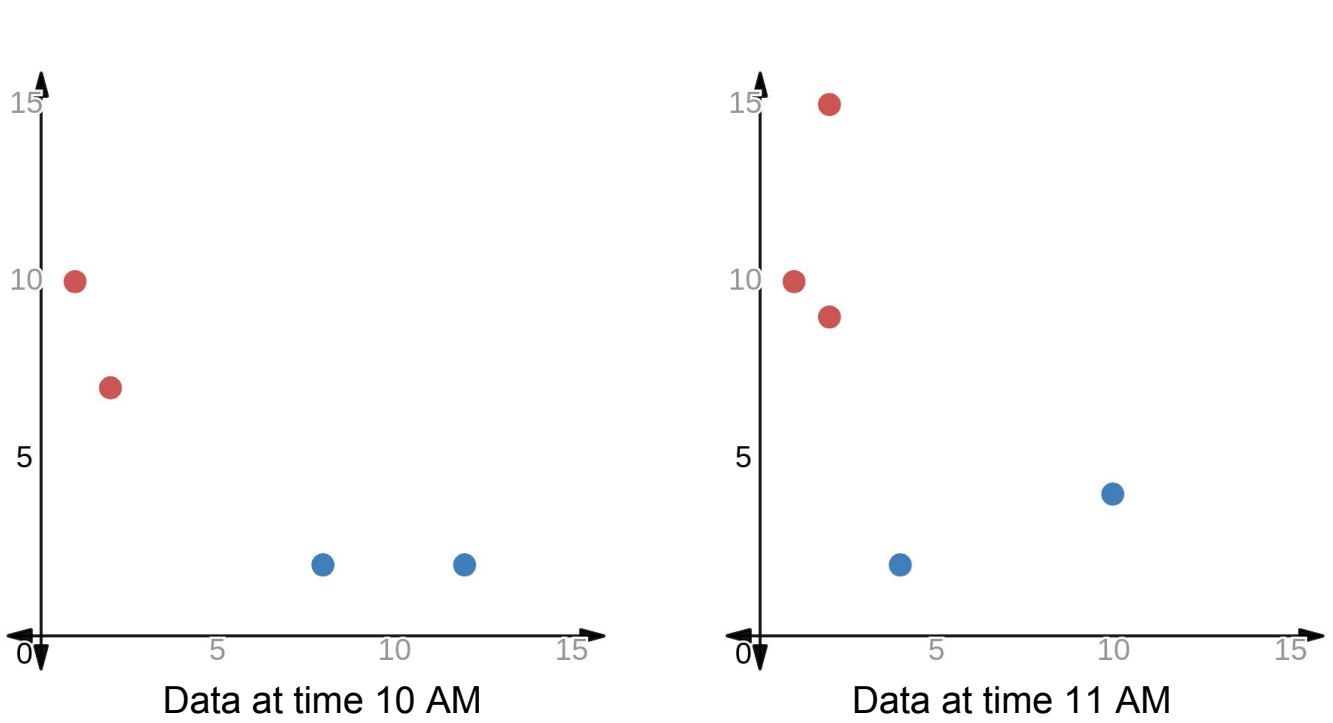
- A novel multi-task learning paradigm that trains multiple soft-margin support vector machine (SVM) classifiers
- Places smooth constraints on the classifiers
- Learns iteratively from a set of related data clusters

new

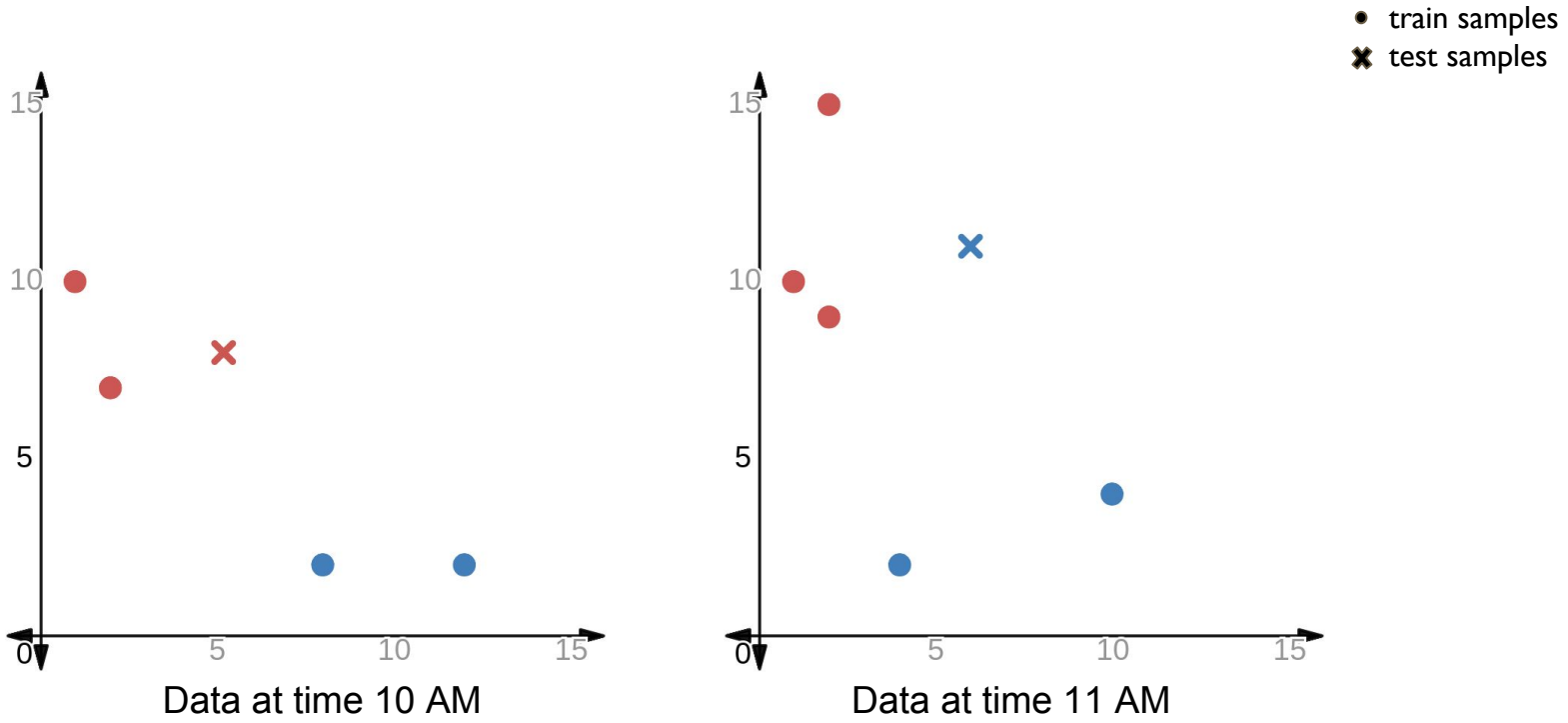
$$\arg\min_{\mathbf{w},b,\xi_i} \frac{1}{4}\|\mathbf{w}\|_2^2 + \frac{1}{4K}\sum_{k=1}^{K}\|\mathbf{w}-\mathbf{v}_k\|_2^2 + C\sum_{i=1}^{n_c}\xi_i$$

$$\text{such that} \quad y_i(b+\mathbf{w}^T\mathbf{x}_i) \geq 1-\xi_i, \xi_i \geq 0, \forall i$$

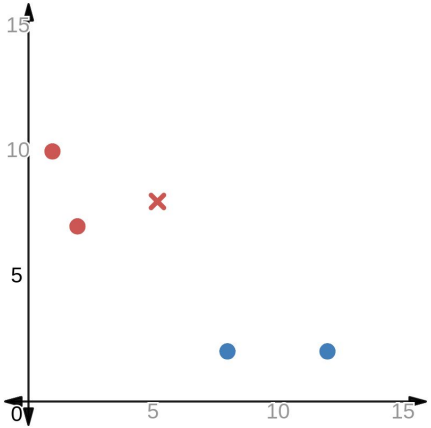
Demonstration of smooth SVM: temporal data example

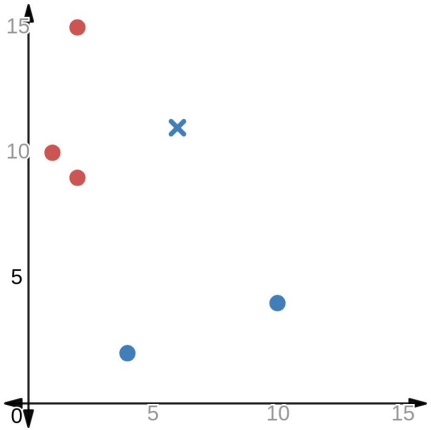


Demonstration of smooth SVM: temporal data example

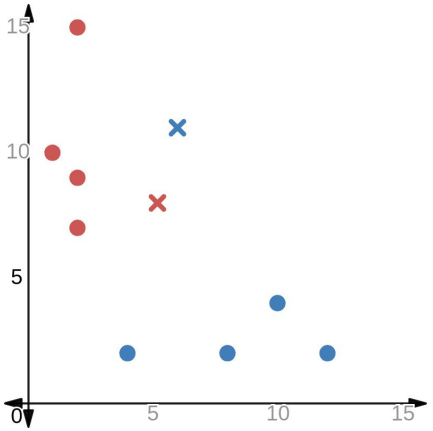


Demonstration of smooth SVM: the spectrum

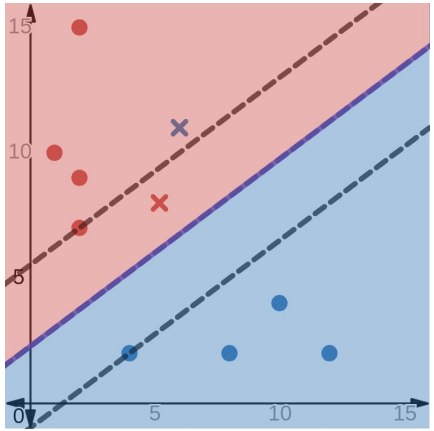




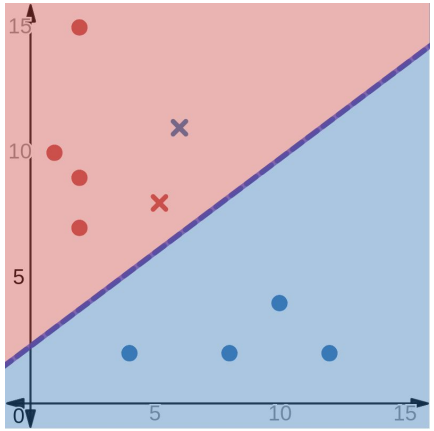
Demonstration of smooth SVM: the spectrum



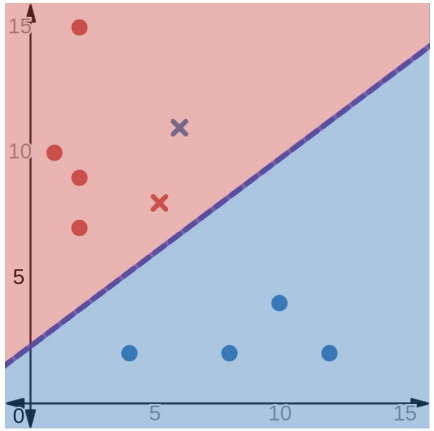
Demonstration of smooth SVM: the spectrum

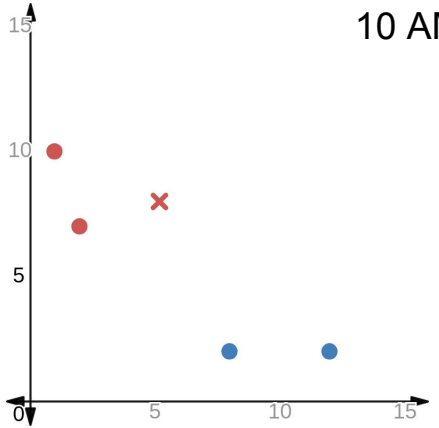


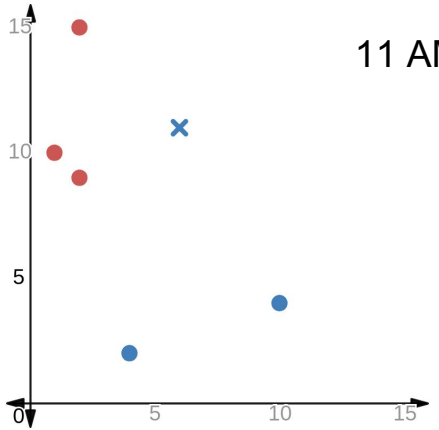
Demonstration of smooth SVM: the spectrum



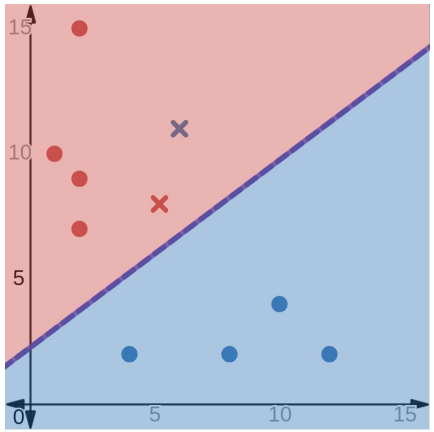
Demonstration of smooth SVM: the spectrum

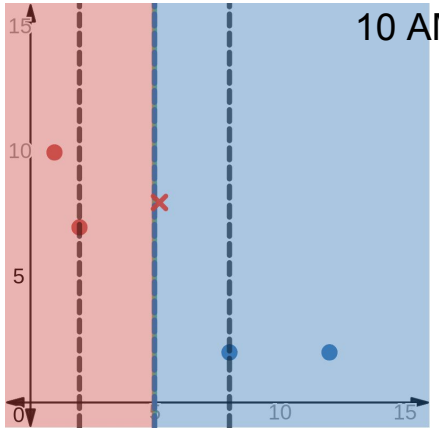


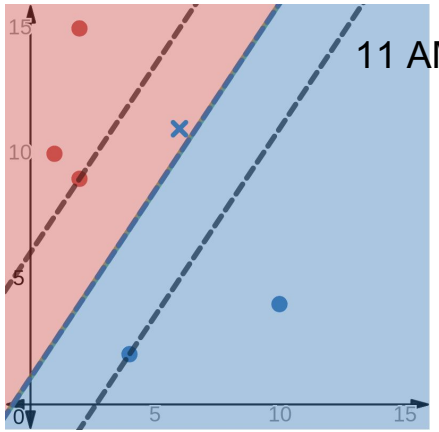




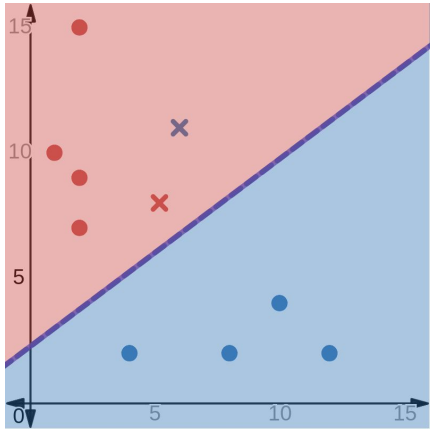
Demonstration of smooth SVM: the spectrum

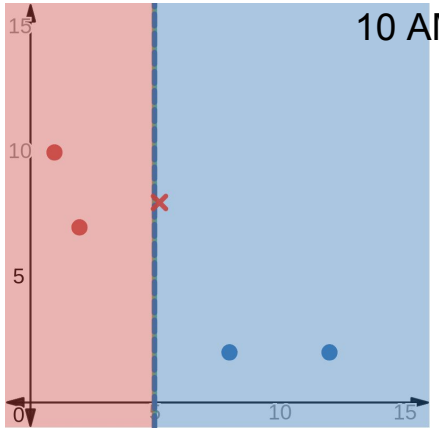


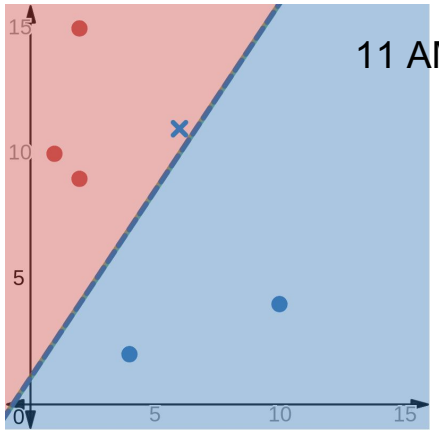




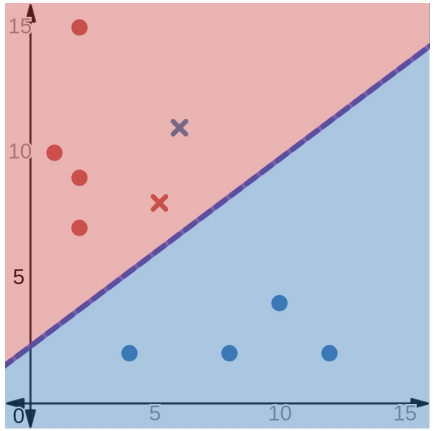
Demonstration of smooth SVM: the spectrum

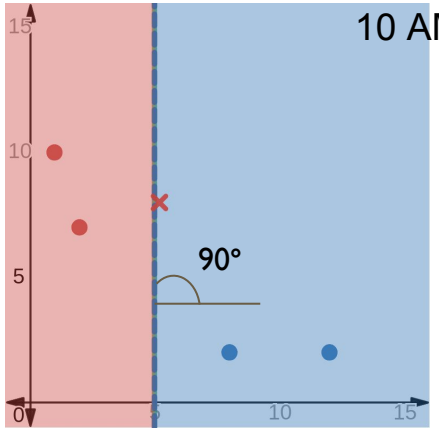


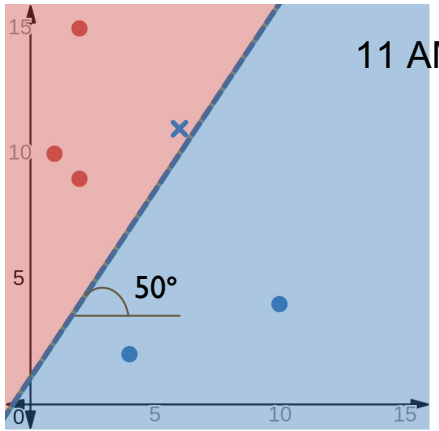




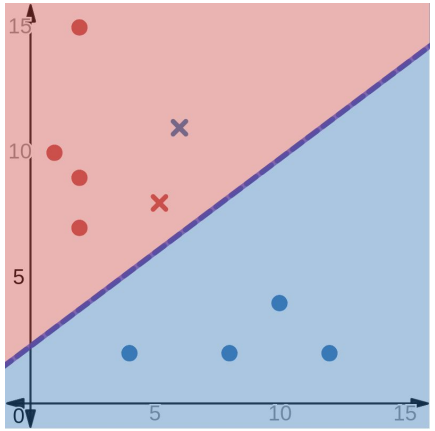
Demonstration of smooth SVM: the spectrum

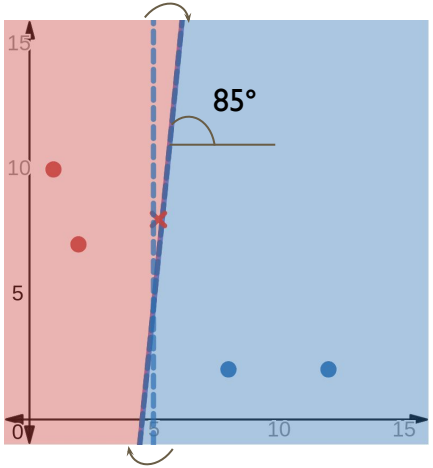


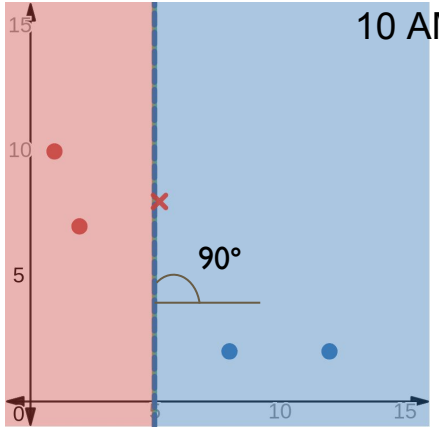


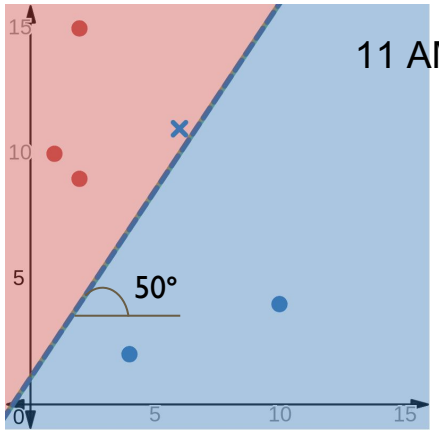


Demonstration of smooth SVM: the spectrum

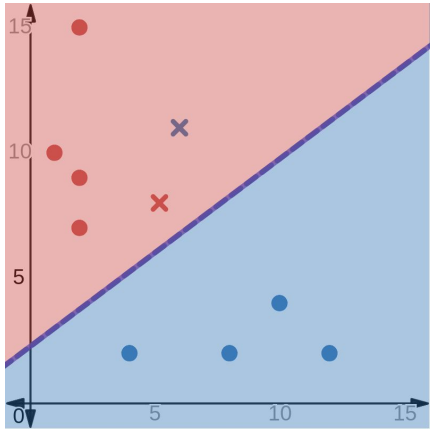


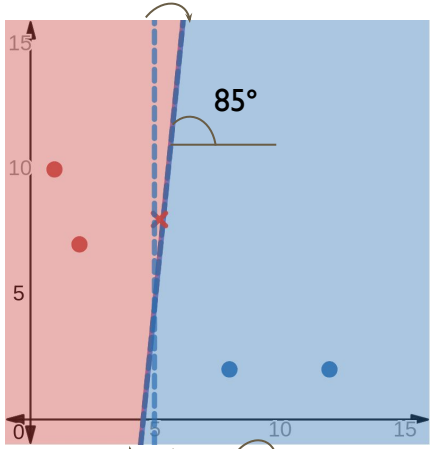


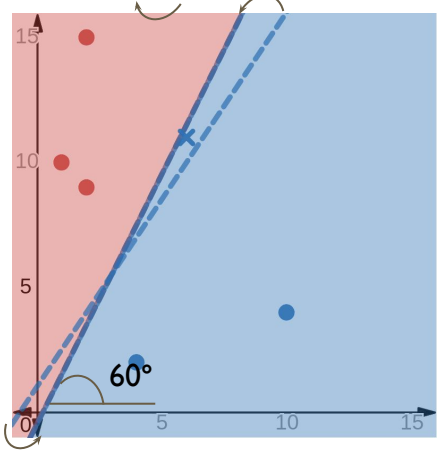


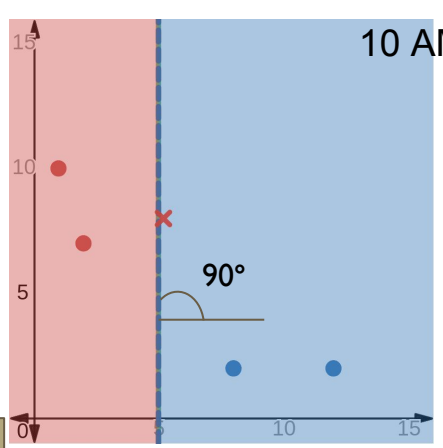


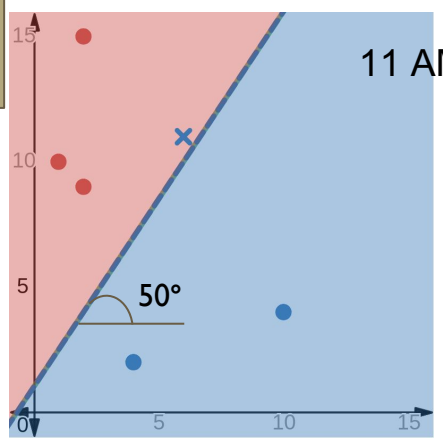
Demonstration of smooth SVM: the spectrum







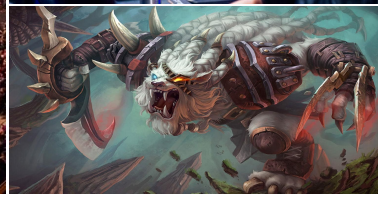




Outline

- Motivation
- Background
- Idea
- Datasets
- Proposed method
- Experimental results

Datasets



League of Legends:

- One of the most played Multiplayer Online Battle Arena games
- Battle between 2 teams of 5 players
- 7 tiers for different levels of expertise

Los Angeles traffic map:

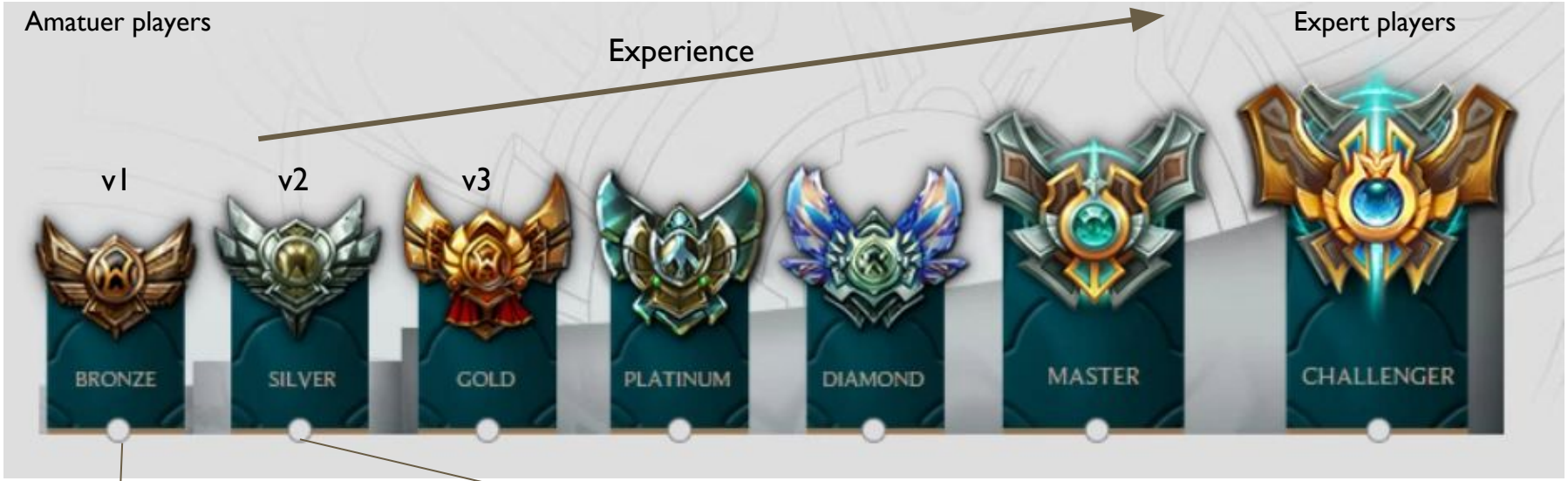
- April 2011, every 5 minutes
- Every day 7-11AM
- 100 road segments.

60% Training Data 122+k games	20% Validation 41+k games	20% Testing 41+k games

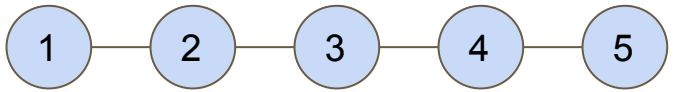
Outline

- Motivation
- Background
- Idea
- Datasets
- Proposed method
- Experimental results

Formulation for games in different levels of difficulties



$$\arg\min P + \|\mathbf{w} - \mathbf{v}_2\|_2^2 \qquad \arg\min P + \|\mathbf{w} - \mathbf{v}_1\|_2^2 + \|\mathbf{w} - \mathbf{v}_3\|_2^2 \qquad \bullet\ \bullet\ \bullet$$



Two methods for optimization

1- Dual form (classical SVM):

$$L_D(\alpha, \mu) = \arg\min_{w,\xi} L(w, \xi, \alpha, \mu)$$

+ computationally efficient
- changes the feature space

2- Primal form (Hinge loss):

$$L_P = \arg\min_{w,b} L(w, b)$$

+ relevant features are extractable
- kernel trick is not applicable

Two methods for optimization

1- Dual form (classical SVM):

$$\arg\min_{\mathbf{w},b,\xi_i} \frac{1}{4}\|\mathbf{w}\|_2^2 + \frac{1}{4K}\sum_{k=1}^{K}\|\mathbf{w}-\mathbf{v}_k\|_2^2 + C\sum_{i=1}^{n_c}\xi_i - \sum_{i=1}^{n_c}\alpha_i(y_i(b+\mathbf{w}^T\mathbf{x}_i)-1+\xi_i) - \sum_i \beta_i\xi_i$$

+ computationally efficient

- changes the feature space

2- Primal form (Hinge loss):

$$\arg\min_{\mathbf{w},b} \sum_{i=1}^{n_c}[1-y_i(b+\mathbf{w}^T\mathbf{x}_i)]_+ + \frac{\lambda_2}{4}\|\mathbf{w}\|_2^2 + \frac{\lambda_2}{4K}\sum_{k=1}^{K}\|\mathbf{w}-\mathbf{v}_k\|_2^2 + \lambda_1\|\mathbf{w}\|_1$$

+ relevant features are extractable

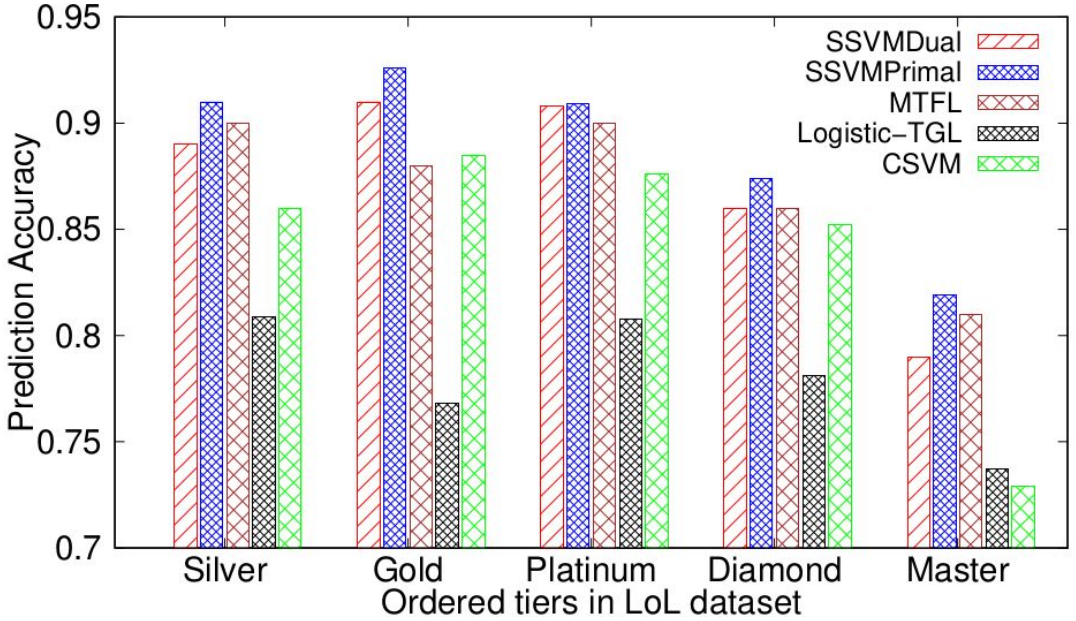
- kernel trick is not applicable

Outline

- Motivation
- Background
- Idea
- Datasets
- Proposed method
- Experimental results

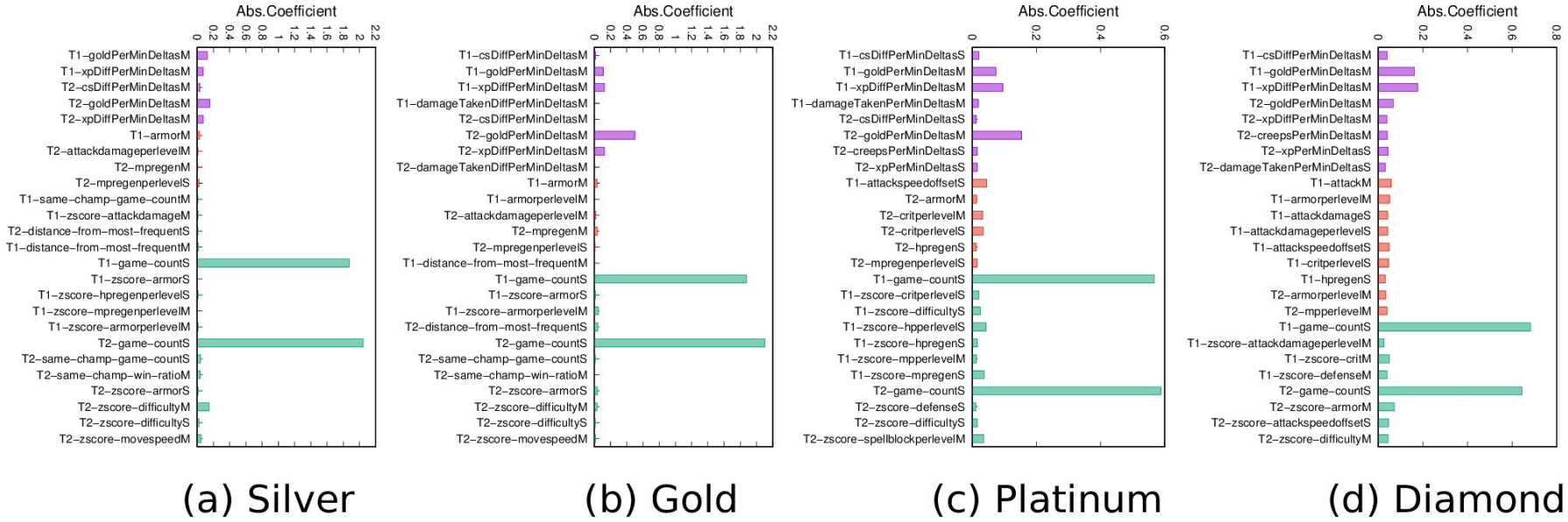
Primal formulation outperforms the baseline methods

- **CSVM:** Classical SVM with no constraint on smoothness
- **MTFL:** Multi-task Feature Learning that uses linear regression [1]
- **Logistic-TGL:** Logistic Multi-task Learning that mostly has studied in disease progression data [2]



[1] Zhao, L., Q. Sun, J. Ye, F. Chen, C. Lu, and N. Ramakrishnan. "Multi-task learning for spatio-temporal event forecasting.", SIGKDD, (2015)
[2] Zhou, J., L. Yuan, J. Liu, and J. Ye. "A multi-task learning formulation for predicting disease progression.", SIGKDD, (2011)

Feature interpretability

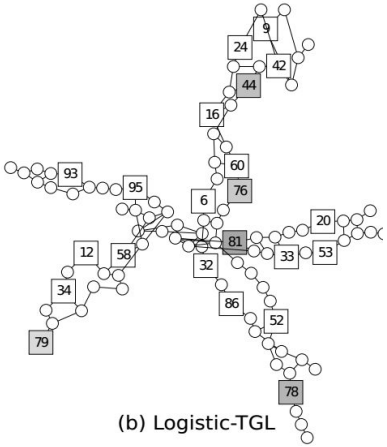


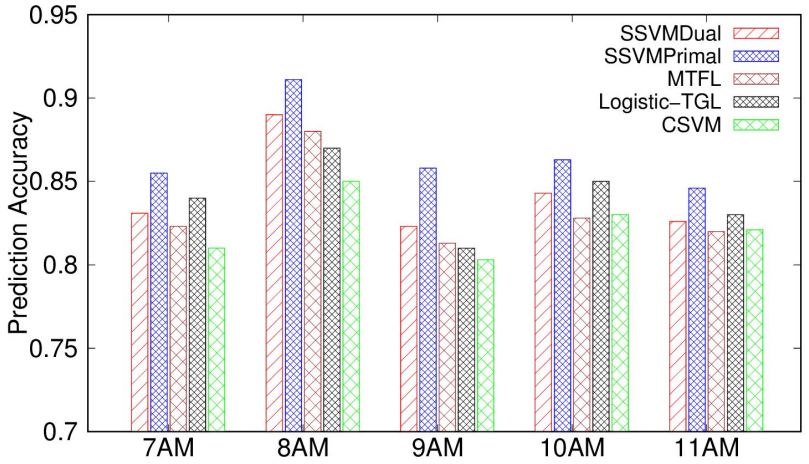
(a) Silver (b) Gold (c) Platinum (d) Diamond

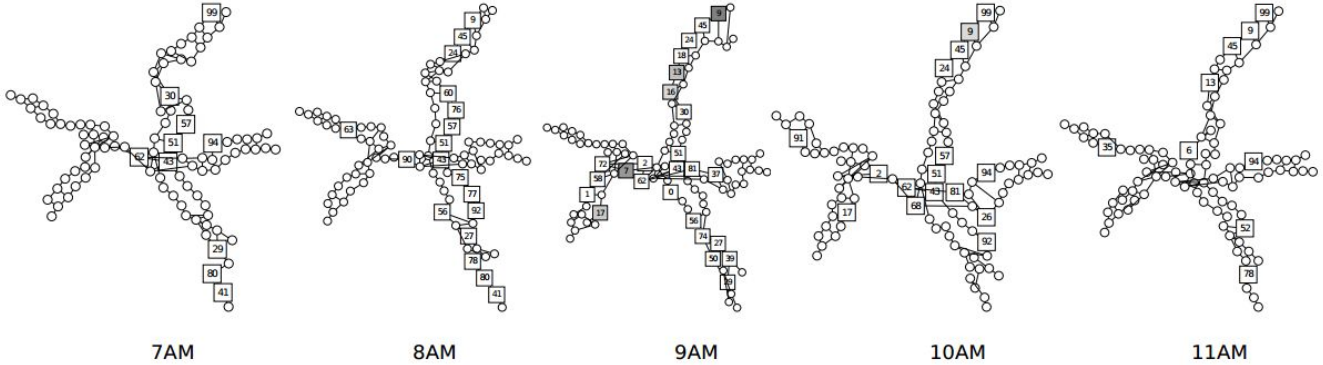
Selected features in SVMPrimal model. Purple encodes team-based features; Orange encodes champions' features; Green encodes individual player proficiency features

In high-tiered games, team features are of more significance compared to individual features.

L.A. traffic dataset







Applying this method on L.A. traffic map depicts the growth/shrink of road congestions in subsequent hours are accurately detected by our model.

Takeaways

- Novel learning framework for multi-task learning with ability to vary distribution of features over different tasks.
- Verification of an interesting result in team formation: in more experienced teams, coordination and team-related features drives the performance, while in less experienced teams, individual skills are deciding the performance.
- Outperforming state-of-the-art methods in multi-task learning in two real datasets:
    - By eliminating irrelevant features, the primal-form SVM is less susceptible to noisy data. Its searching space is also much smaller/narrower so it can converge to better classification boundaries.

Future works:

- Applying the method on any data with hierarchical clustering
- Introducing nonlinearity to the model (i.e. kernel) in sacrifice of interpretability

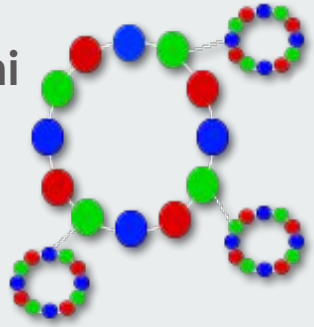
Thanks to our generous sponsors

- Research was supported by the U.S. Army Research Laboratory and the U.S. Army Research Office under grant number W911NF-15-1-0577, Network Science of Teams, Multidisciplinary University Research Initiative (MURI) award.

Learning Multiclassifiers with Predictive Features that Vary with Data Distribution

Omid Askarisichani
Xuan-Hong Dang
Ambuj Singh

13 December 2018



Back up slides

Dual form

Lagrangian dual objective function:

$$\arg\min_{\mathbf{w},b,\xi_i} \frac{1}{4}\|\mathbf{w}\|_2^2 + \frac{1}{4K}\sum_{k=1}^{K}\|\mathbf{w}-\mathbf{v}_k\|_2^2 + C\sum_{i=1}^{n_c}\xi_i - \sum_{i=1}^{n_c}\alpha_i(y_i(b+\mathbf{w}^T\mathbf{x}_i)-1+\xi_i) - \sum_i \beta_i\xi_i$$

smoothness

$$\frac{\partial L}{\partial \mathbf{w}} = \frac{1}{2}\mathbf{w} + \frac{1}{2K}\sum_{k=1}^{K}(\mathbf{w}-\mathbf{v}_k) - \sum_{i=1}^{n_c}\alpha_i y_i \mathbf{x}_i = 0 \qquad \frac{\partial L}{\partial b} = \sum_{i=1}^{n_c}\alpha_i y_i = 0 \qquad \frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0$$

Plugging these into the objective function:

$$\arg\min_{\alpha} \sum_{i=1}^{n_c}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j x_i^T y_i^T y_j x_j - \frac{1}{2K}\Big(\sum_{k=1}^{K}\mathbf{v}_k\Big)^T \sum_{i=1}^{n_c}\alpha_i y_i x_i$$

$$\text{such that} \quad 0 \le \alpha_i \le C \text{ and } \sum_{i=1}^{n_c}\alpha_i y_i = 0$$

Quadratic programing finds the support vectors.

Primal form

Objective function conditions can be formatted as:

$$\begin{matrix} y_i(b + \mathbf{w}^T\mathbf{x}_i) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{matrix}$$

hinge loss function

$$\xi_i \geq \max(0, 1 - y_i(b + \mathbf{w}^T\mathbf{x}_i)) = [1 - y_i(b + \mathbf{w}^T\mathbf{x}_i)]_+$$

Thus, the objective function in primal form:

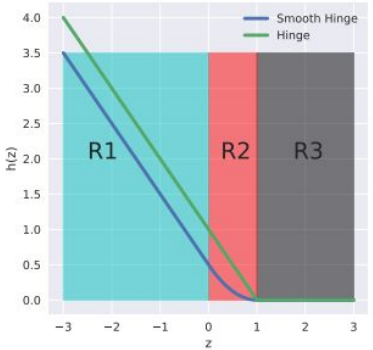
smoothness sparsity

$$\arg\min_{\mathbf{w},\mathbf{b}} \sum_{i=1}^{n_c} h\Big(y(b + w^T x_i)\Big) + \frac{\lambda_2}{4}\|\mathbf{w}\|_2^2 + \frac{\lambda_2}{4K}\sum_{k=1}^{K}\|\mathbf{w} - \mathbf{v}_k\|_2^2 + \lambda_1\|\mathbf{w}\|_1$$

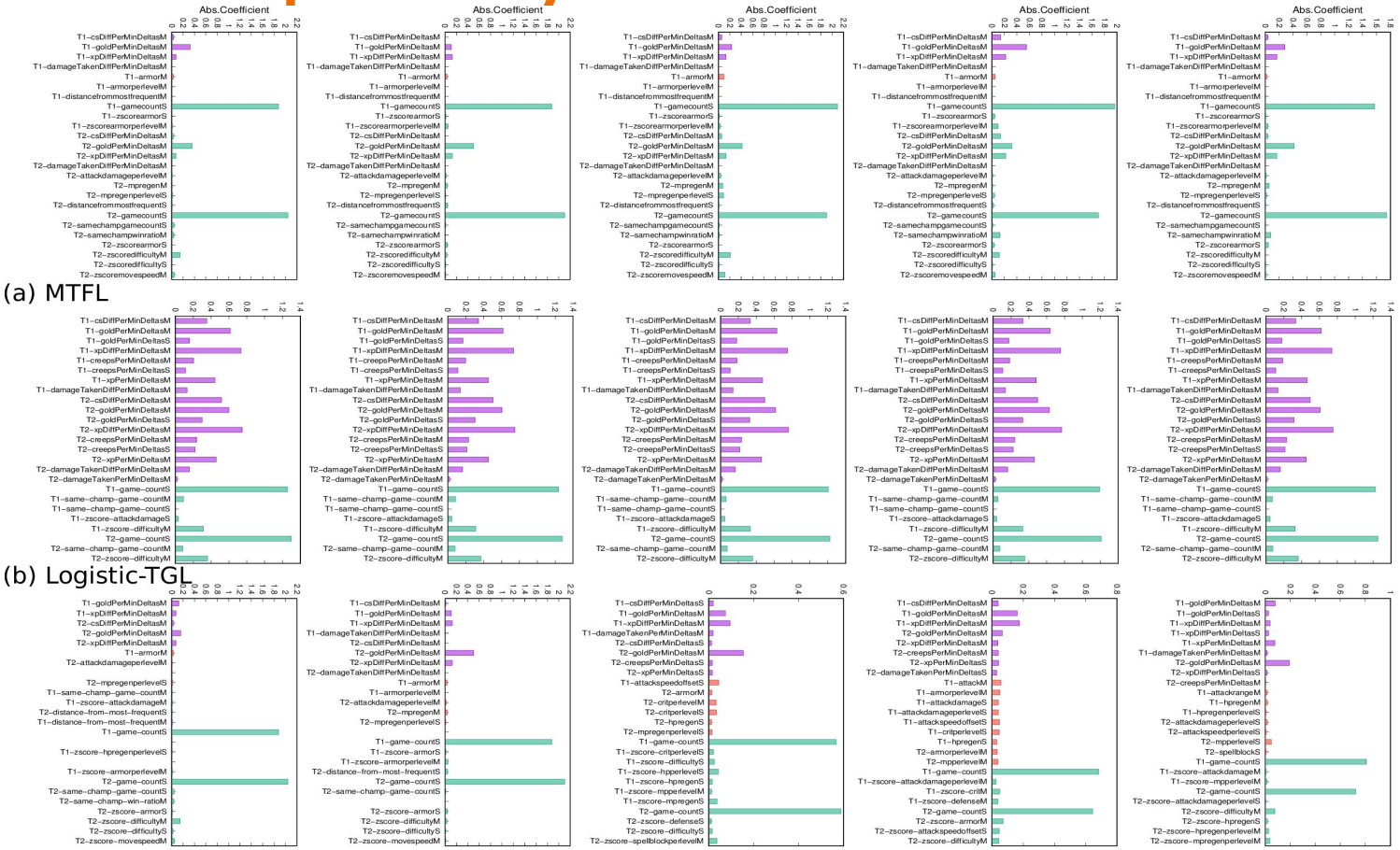
It can be directly optimized in single update for each parameter:

$$b = |R_2|^{-1}\left(\sum_{z_i \in R_1} y_i - \sum_{z_i \in R_2} (\mathbf{w}^T\mathbf{x}_i - y_i)\right)$$

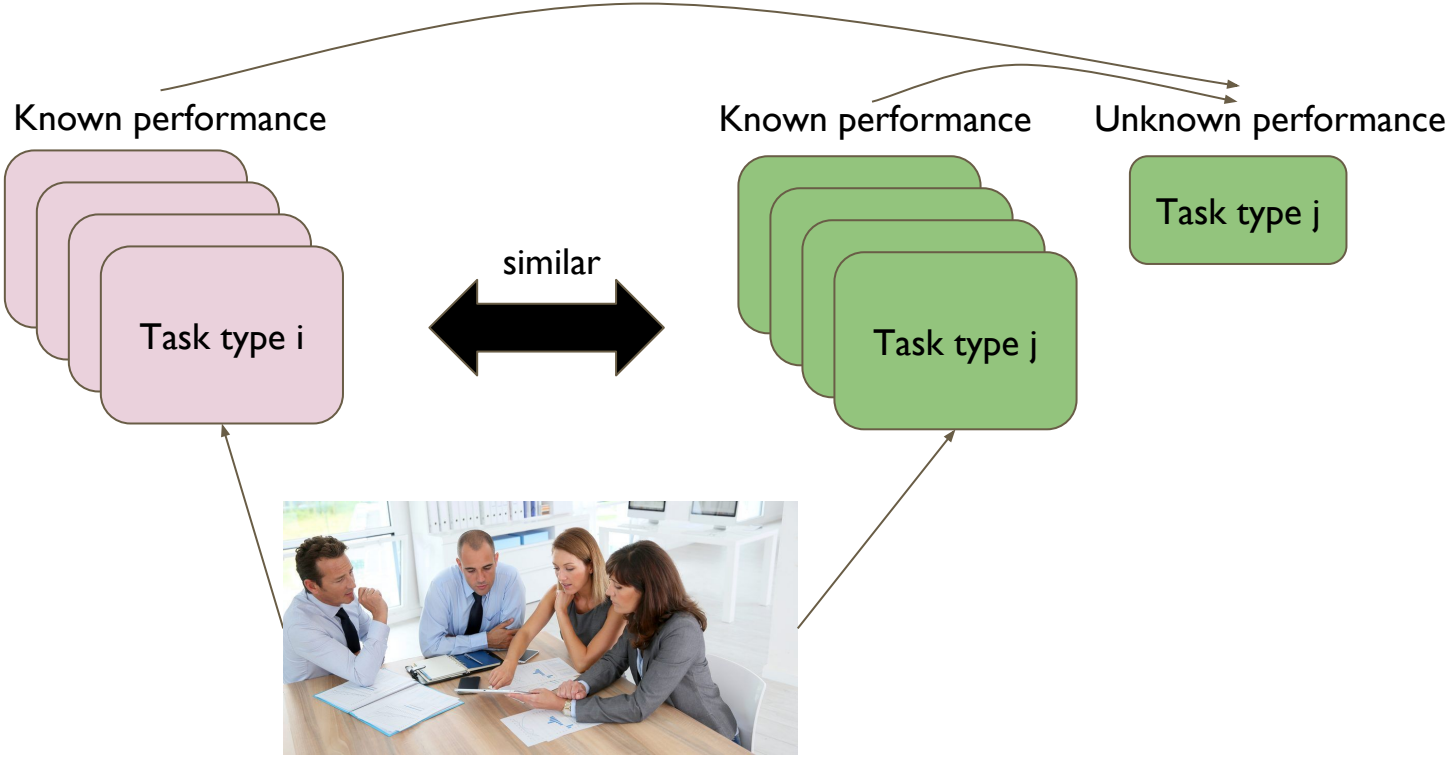
$$w_j = \begin{cases} (c + \lambda_1)/a & \text{if } c < -\lambda_1 \\ (c - \lambda_1)/a & \text{if } c > +\lambda_1 \\ 0 & \text{if } c \in [-\lambda_1; +\lambda_1] \end{cases}$$



Feature interpretability



Motivation: team performance on similar tasks



Smooth SVM example