Distance Based SVM Kernels for Infinite Ensemble Learning

Hsuan-Tien Lin and Ling Li

Learning Systems Group, California Institute of Technology

ICONIP, November 2, 2005

Setup

- notation:
  - examples: $x \in \mathcal{X} \subseteq \mathbb{R}^D$
  - labels: $y \in \{+1, -1\}$
  - hypotheses (classifiers): functions from $\mathcal{X} \to \{+1, -1\}$
- binary classification problem: given training examples and labels $\{(x_i, y_i)\}_{i=1}^N$, find a classifier $g(x)$ that predicts the labels of unseen examples well
- ensemble learning: weighted vote of a committee of hypotheses

$$g(x) = \text{sign}\left(\sum w_t h_t(x)\right), w_t \geq 0, h_t \in \mathcal{H}$$

$g(\cdot)$ is usually better than individual $h(\cdot)$

Traditional Ensemble Learning

- traditional ensemble learning: iteratively find (w_t, h_t) for $t = 1, 2, \cdots, T$

$$g(x) = \text{sign}\left(\sum_{t=1}^{T} w_t h_t(x)\right), w_t \geq 0, h_t \in \mathcal{H}$$

- AdaBoost: asymp. approximate an optimal ensemble

$$\min_{w,h} \|w\|_1, \text{ s.t. } y_i \left(\sum_{t=1}^{\infty} w_t h_t(x_i)\right) \geq 1, w_t \geq 0$$

by T iterations of coordinate descent on barrier algorithm

is an infinite ensemble better?

Infinite Ensemble Learning

- infinite ensemble learning: $|\mathcal{H}| = \infty$, and possibly **infinite** number of nonzero weights w_t

$$g(x) = \text{sign}\left(\sum w_t h_t(x)\right), h_t \in \mathcal{H}, w_t \geq 0$$

- infinite ensemble learning is a challenge (e.g. Vapnik, 1998)
- SVM can handle infinite number of weights with suitable kernels (e.g. Schölkopf and Smola, 2002)
- SVM and AdaBoost are connected (e.g. Rätsch et al., 2001)

can SVM be applied to infinite ensemble learning?

Connection between SVM and AdaBoost

SVM	AdaBoost
$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\lvert\xi_i\rvert$ $y_i\left(\sum_{d=1}^{\infty} w_d\phi_d(x_i) + b\right) \geq 1 - \xi_i$	$\min_{w,h} \|w\|_1$ $y_i\left(\sum_{t=1}^{\infty} w_t h_t(x_i)\right) \geq 1$ $w_t \geq 0$
optimization	
dual solution with quadratic programming	asymp. approximate with barrier and coordinate descent
key for infinity	
kernel trick $\mathcal{K}(x, x') = \sum \phi_d(x)\phi_d(x')$	approximation

$$\boldsymbol{\phi_d(x) \Longleftrightarrow h_t(x)}$$

Framework of Infinite Ensemble Learning

Algorithm

1. Consider a hypothesis set $\mathcal{H}$
2. Embed $\mathcal{H}$ in a kernel $\mathcal{K}_{\mathcal{H}}$ using $\phi_d \Leftrightarrow h_t$
3. Properly choose other SVM parameters
4. Train SVM with $\mathcal{K}_{\mathcal{H},r}$ and $\{(x_i, y_i)\}_{i=1}^{N}$
5. Output an infinite ensemble classifier

- If $\mathcal{H}$ is negation complete and contains a constant hypothesis, SVM classifier is equivalent to an infinite ensemble classifier
- SVM as an optimization machinery: training routines are widely available (LIBSVM)
- SVM as a well-studied learning model: inherit the profound regularization properties

hard part: kernel construction

Embedding Hypotheses into the Kernel

$$\mathcal{K}(x, x') = \sum_{t=1}^{\infty} h_t(x) h_t(x') \quad \text{(may not converge)}$$

$$\Rightarrow \quad \mathcal{K}(x, x') = \sum_{d=1}^{\infty} [r_d h_d(x)][r_d h_d(x')] \quad \text{(with some positive } r\text{)}$$

$$\Rightarrow \quad \mathcal{K}(x, x') = \int [r(\alpha) h_\alpha(x)][r(\alpha) h_\alpha(x')] \, d\alpha \quad \text{(handle uncountable cases)}$$

- Let $\phi_x(\alpha) = r(\alpha) h_\alpha(x)$, the kernel $\mathcal{K}_{\mathcal{H},r}(x, x') = \int_{\mathcal{C}} \phi_x(\alpha) \phi_{x'}(\alpha) \, d\alpha$ embodies $\mathcal{H} = \{h_\alpha : \alpha \in \mathcal{C}\}$
- $\mathcal{K}_{\mathcal{H},r}(x, x')$: an inner product for ϕ_x and $\phi_{x'}$ in $\mathcal{L}_2(\mathcal{C})$

examples: stump and perceptron kernels

Stump Kernel

- decision stump: $s_{q,d,\alpha}(x) = q \cdot \text{sign}((x)_d - \alpha)$
- simplicity: popular for ensemble learning
- consider $\mathcal{S} = \left\{s_{q,d,\alpha_d} : q \in \pm 1, 1 \le d \le D, \alpha_d \in [L_d, R_d]\right\}$:

Definition

The stump kernel $\mathcal{K}_{\mathcal{S}}$ is defined for $\mathcal{S}$ with $r(q, d, \alpha_d) = \frac{1}{2}$:

$$\mathcal{K}_{\mathcal{S}}(x, x') = \Delta_{\mathcal{S}} - \|x - x'\|_1,$$

where $\Delta_{\mathcal{S}} = \frac{1}{2}\sum_{d=1}^{D}(R_d - L_d)$ is a constant

Perceptron Kernel

- a simple hyperplane: $p_{\theta,\alpha}(x) = \text{sign}\left(\theta^T x - \alpha\right)$
- not easy for ensemble learning: hard to design good algorithm
- consider $\mathcal{P} = \{p_{\theta,\alpha} \colon \|\theta\|_2 = 1, \alpha \in [-R, R]\}$:

Definition

The perceptron kernel $\mathcal{K}_{\mathcal{P}}$ is defined for $\mathcal{P}$ with a constant $r(\theta, \alpha)$:

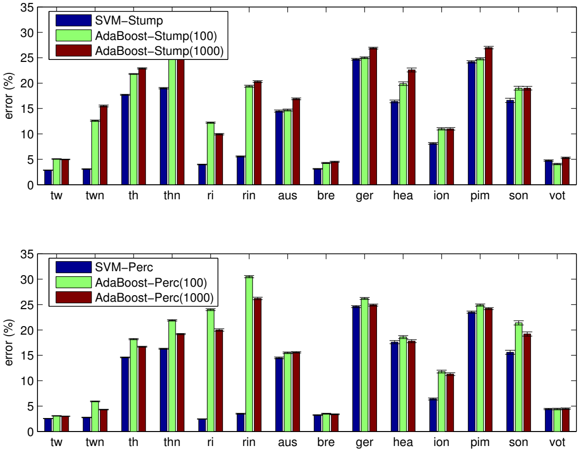
$$\mathcal{K}_{\mathcal{P}}(x, x') = \Delta_{\mathcal{P}} - \|x - x'\|_2,$$

where $\Delta_{\mathcal{P}}$ is a constant.

Properties of the Novel Kernels

- simple to compute: can even drop $\Delta_{\mathcal{S}}$ or $\Delta_{\mathcal{P}}$
  - adding/subtracting a constant does not change the solution under SVM linear constraint $\sum y_i \lambda_i = 0$
- infinite power: perfect separability under mild assumptions
  - similar power to popular Gaussian kernel: $\exp(-\gamma \|x - x'\|_2^2)$
    – suitable control on the power may give good performance
- fast automatic parameter selection: a good parameter C only
  - Gaussian kernel depends on a good (γ, C) pair
    (usually 10 times more computation)
- feature space interpretation: domain-specific tuning
  - e.g. stump kernel for gene selection:
    the stump weight is a natural estimate of gene importance
    (Lin and Li, ECML/PKDD Discovery Challenge, 2005)

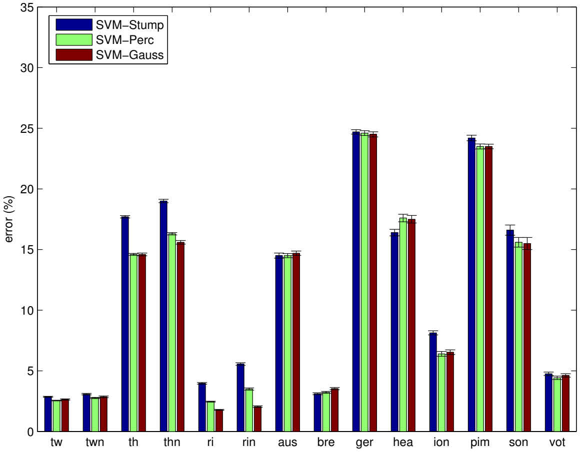
Comparison between SVM and AdaBoost



Results

- fair comparison between AdaBoost and SVM
- SVM is usually best – benefits to go to infinity

Comparison of SVM Kernels



Results

- SVM-Perc very similar to SVM-Gauss
- SVM-Stump comparable to, but sometimes a bit worse than others

Conclusion

- derived two useful kernels: stump kernel and perceptron kernel
- provided meanings to specific distance metric
- stump kernel: succeeded in specific applications
  – existing AdaBoost-Stump applications may switch
- perceptron kernel: similar to Gaussian, faster in parameter selection
  – can be an alternative to SVM-Gauss
- not the only kernels:
  - Laplacian kernel $\rightarrow$ infinite ensemble of decision trees
  - Exponential kernel $\rightarrow$ infinite ensemble of decision regions
- SVM: a machinery for conquering infinity
  – possible to apply similar machinery to areas that need infinite or lots of aggregation