Learning Sequence Kernels

Afshin Rostamizadeh
NYU/Google
rostami@cs.nyu.edu

Joint work with
Corinna Cortes (Google) and Mehryar Mohri (NYU/Google)

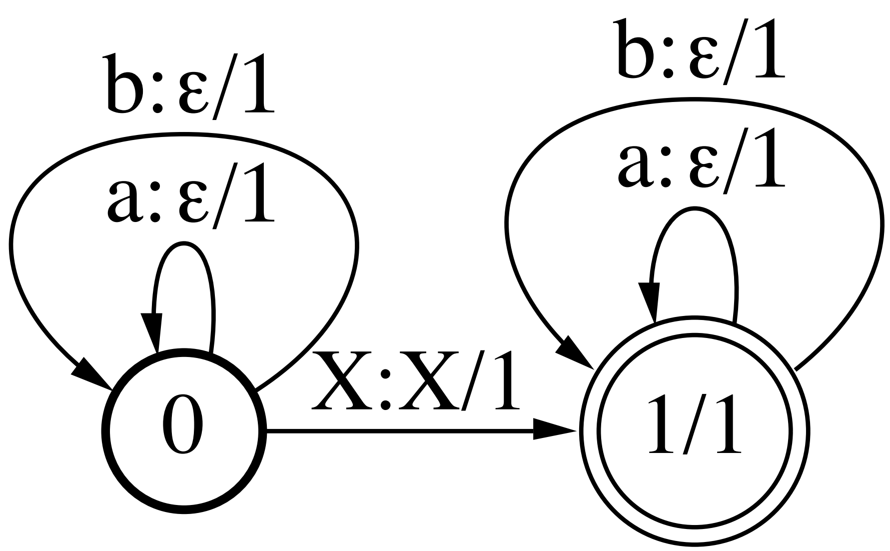
Motivation

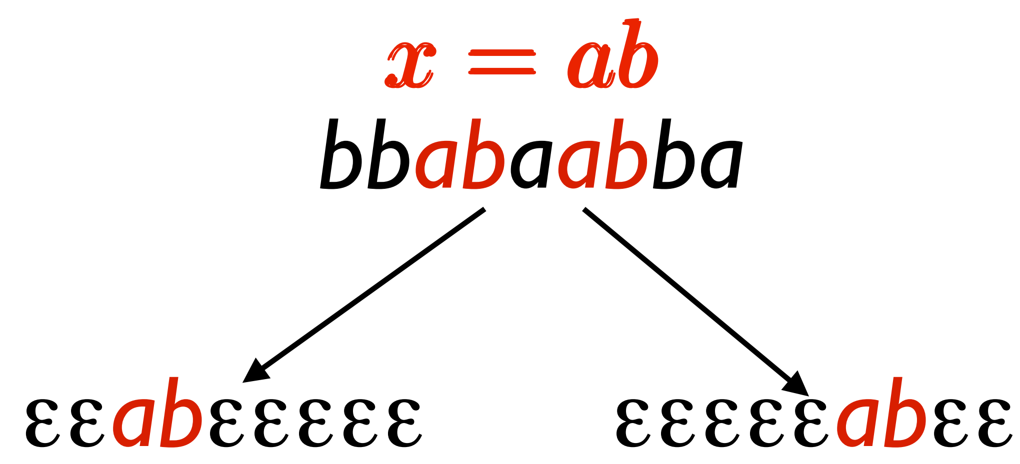
- Kernel methods: widely and successfully used in ML.
- Key component: definition of kernel.
- Arbitrary kernel: any PDS kernel can be used.
- But, the choice is critical to the success: poor selections may lead to sub-optimal performances.
- Instead: use sample points to learn the kernel.
- How do we learn efficiently kernels for sequence data?

Previous Work

- (Lanckriet et al., 2004):
  - learning kernel matrix; transductive setting.
  - SDP formulation; interior point method (Kim et al., 2008).
- (Ong, Smola, Williamson, 2005):
  - kernel function, *hyperkernels*, convex combinations of infinitely many kernels, SDP formulation.
- (Miccheli and Pontil, 2005; Argyriou, Miccheli and Pontil, 2005):
  - kernel function.
  - DC program (difference of convex functions).

Counting Transducers





- X is an automaton representing a string or any other regular expression.
- Alphabet $\Sigma = \{a, b\}$.

SVM Kernel Learning Formulation

$$\min_{K \in \mathcal{K}} \max_{\boldsymbol{\alpha}} \quad 2\boldsymbol{\alpha}^\top \mathbf{1} - \boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{K} \mathbf{Y} \boldsymbol{\alpha}$$

$$\text{subject to} \quad \boldsymbol{\alpha}^\top \mathbf{y} = 0 \ \wedge \ \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C}$$

$$K \succeq 0 \ \wedge \ \mathrm{Tr}[\mathbf{K}] = \Lambda,$$

where $\Lambda > 0$ determines the family of kernels.

Structural Risk Minimization (SRM)

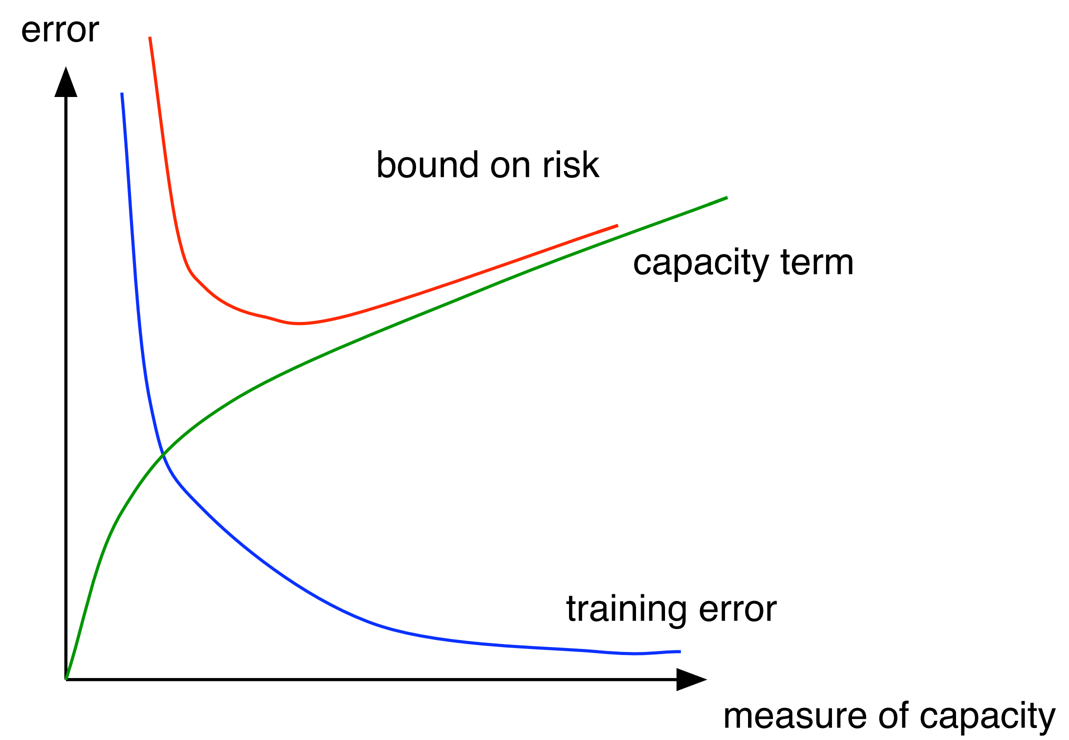
(Vapnik, 1995)

- Principle: consider an infinite sequence of hypothesis spaces ordered for inclusion:

$$H_1 \subset H_2 \subset \cdots \subset H_n \cdots$$

Then, select hypothesis *h* minimizing the trade-off:

$$h = \underset{h \in H_n, n \in \mathbb{N}}{\operatorname{argmin}} \ \widehat{error}(h) + \text{capacity-measure}(H_n, h).$$



Count-Based Kernels

$$K(x_i, x_j) = \sum_{k=1}^{p} T(x_i, z_k) T(x_j, z_k)$$
$$= \sum_{k=1}^{p} w_k^2 \, |x_i|_k \, |x_j|_k .$$

Kernel matrix:

$$\mathbf{K} = \sum_{k=1}^{p} \mu_k \mathbf{X}_k \mathbf{X}_k^\top, \text{ with } \begin{array}{l} \mu_k = w_k^2 \\ \mathbf{X}_{ik} = |x_i|_k . \end{array}$$

SVM - Dual Optimization Problem

$$\min_{\boldsymbol{\mu}} \max_{\boldsymbol{\alpha}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}) = 2\boldsymbol{\alpha}^\top \mathbf{1} - \sum_{k=1}^{p} \mu_k \boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{Y} \boldsymbol{\alpha}$$

$$\text{subject to} \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C} \wedge \boldsymbol{\alpha}^\top \mathbf{y} = 0$$

$$\boldsymbol{\mu} \geq \mathbf{0} \wedge \sum_{k=1}^{p} \mu_k \|\mathbf{X}_k\|^2 = \Lambda.$$

Minimax Property

- By von Neumann's generalized minmax theorem:

$$\min_{\boldsymbol{\mu}\in\mathcal{M}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}\in\mathcal{A}} \min_{\boldsymbol{\mu}\in\mathcal{M}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}).$$

- max-min optimization:

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}} \min_{\boldsymbol{\mu}\in\mathcal{M}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}\in\mathcal{A}} 2\boldsymbol{\alpha}^\top \mathbf{1} - \max_{\boldsymbol{\mu}\in\mathcal{M}} \sum_{k=1}^{p} \mu_k (\boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{X}_k)^2.$$

Simplification

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}} 2\boldsymbol{\alpha}^\top \mathbf{1} - \max_{\boldsymbol{\mu}\in\mathcal{M}} \sum_{k=1}^{p} \mu_k (\boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{X}_k)^2$$

$$= \max_{\boldsymbol{\alpha}\in\mathcal{A}}\ 2\boldsymbol{\alpha}^\top \mathbf{1} - \Lambda \max_{k\in[1,p]} \left(\frac{\boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{X}_k}{\|\mathbf{X}_k\|} \right)^2$$

$$= \max_{\boldsymbol{\alpha}\in\mathcal{A}}\ 2\boldsymbol{\alpha}^\top \mathbf{1} - \Lambda \max_{k\in[1,p]} (\boldsymbol{\alpha}^\top \mathbf{u}'_k)^2,$$

with $\mathbf{u}'_k = \frac{\mathbf{Y}^\top \mathbf{X}_k}{\|\mathbf{X}_k\|} = \frac{\mathbf{Y}^\top \mathbf{X}_k}{\|\mathbf{Y}^\top \mathbf{X}_k\|}$.

SVM - QP Formulation

$$\min_{\boldsymbol{\alpha}, t} \quad -2\boldsymbol{\alpha}^\top \mathbf{1} + \Lambda t^2$$

$$\text{subject to} \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C} \wedge \boldsymbol{\alpha}^\top \mathbf{y} = 0$$

$$-t \leq \boldsymbol{\alpha}^\top \mathbf{u}'_k \leq t, \forall k \in [1, p].$$

Where $\mathbf{u}'_k = \dfrac{\mathbf{Y}^\top \mathbf{X}_k}{\|\mathbf{X}_k\|}.$

SVM - Retrieving μ

$$\min_{\boldsymbol{\alpha},t} \quad -2\boldsymbol{\alpha}^\top \mathbf{1} + \Lambda t^2$$

$$\text{subject to} \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C} \wedge \boldsymbol{\alpha}^\top \mathbf{y} = 0$$

$$\underbrace{-t \leq}_{\beta'_k} \boldsymbol{\alpha}^\top \mathbf{u}'_k \underbrace{\leq t,}_{\beta_k} \forall k \in [1,p].$$

$$\mu_k = \frac{\beta_k + \beta'_k}{2t\|X_k\|^2}.$$

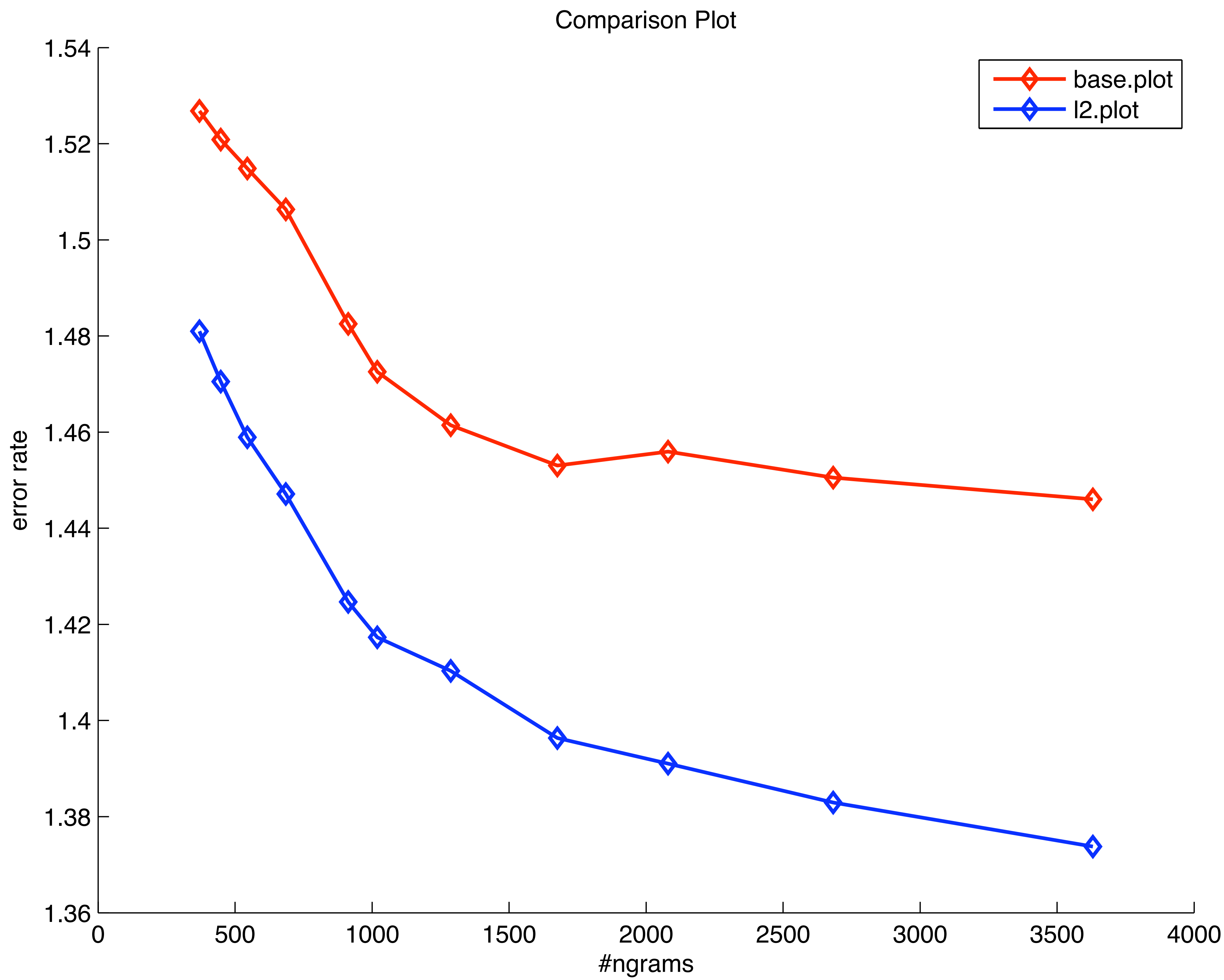
Experiments

- Dataset:
  - Regression: sentiment analysis dataset.
    - 2,000 data points.
    - Bigram count features.
- Set-up:
  - Baseline: bigram kernel with uniform weights.
  - 10-fold cross-validation.
  - Increasing number of bigrams.

Feature Selection

- Kitchen appliances:
  - Gives large weights to discriminative features:
    - great_little, great_product, is_perfect, are_great, and_looks, beautiful_and, ...
    - a_shame, doesn't_work, very_poor, return_it, way_too, very_disappointed, after_just, bother_with, ...
  - Zero weight to many features (L1-regularization encourages sparse solution).

L2 Regularization



Conclusion

- Efficient algorithms for learning count-based sequence kernels (QP formulation, iterative method).
- Learning kernels effective based on empirical evidence.
- How do we learn more complex rational kernels?
- How do we scale algorithms to even larger data sets?