LEARNING FROM DISTRIBUTIONS VIA SUPPORT MEASURE MACHINES

K. Muandet, K. Fukumizu, F. Dinuzzo, B. Schölkopf

Potential Applications:

- Uncertain/noisy data (astronomical/biological data)
- Groups of samples (group anomaly, preference learning)
- Changing environments (domain adaptation/generalization)
- Large-scale machine learning (data squashing)
Regularization on Probability Measures

Given a sample \((P_1, y_1), (P_2, y_2), \ldots, (P_m, y_m)\), any solution \(f\) to
\[
\ell(P_1, y_1, \mathbb{E}_{P_1}[f], \ldots, P_m, y_m, \mathbb{E}_{P_m}[f]) + \Omega \left( \|f\|_H \right) \quad (1)
\]
adopts a form \(f = \sum_{i=1}^{m} \alpha_i \mathbb{E}_{P_i}[k(x, \cdot)]\) for some \(\alpha_i \in \mathbb{R}, i = 1, \ldots, m\).

Our framework (1) is different from
1. \(\mathbb{E}_{P_1} \mathbb{E}_{P_2} \ldots \mathbb{E}_{P_m} \ell(\{x_i, y_i, f(x_i)\}_{i=1}^{m}) + \Omega(\|f\|_H)\) intracable
2. \(\ell(\{M_i, y_i, f(M_i)\}_{i=1}^{m}) + \Omega(\|f\|_H), M_i = \mathbb{E}_{P_i}[x]\) information loss

Risk Deviation Bound
Given a distribution \(\mathbb{P}\) with variance \(\sigma^2\), a Lipschitz continuous function \(f\) with constant \(C_f\), a loss function \(\ell\) with constant \(C_\ell\), it follows for any \(y \in \mathbb{R}\) that
\[
|\mathbb{E}_{X \sim \mathbb{P}}[\ell(y, f(x))] - \ell(y, \mathbb{E}_{X \sim \mathbb{P}}[f(x)])| \leq 2C_\ell C_f \sigma
\]
Information preserving + computationally efficient.
Support Measure Machines (SMM)

Embedding kernel

$$\psi(\mu_P)$$

Level–2 kernel

$$\mu_P = \mathbb{E}_P[k(x, \cdot)]$$

Feature maps

$$K(\delta_x, \delta_y) = \langle k(x, \cdot), k(y, \cdot) \rangle_{\mathcal{H}} = k(x, y)$$

The SVM is recovered as a special case.

Linear kernels

$$K(P, Q) = \langle \mu_P, \mu_Q \rangle_{\mathcal{H}} = \mathbb{E}_{x \sim P, z \sim Q}[k(x, z)]$$

It defines the feature for distributions.

Nonlinear kernels

$$K(P, Q) = \kappa(\mu_P, \mu_Q) = \langle \psi(\mu_P), \psi(\mu_Q) \rangle_{\mathcal{F}}$$

It allows for nonlinear learning algorithms.
Flexible Support Vector Machines

\[ K(\mathbb{P}, \mathbb{Q}) = \left\langle \int k(\tilde{x}, \cdot)g(x, \tilde{x})d\tilde{x}, \int k(\tilde{z}, \cdot)g(z, \tilde{z})d\tilde{z} \right\rangle_{\mathcal{H}} = k_g(x, z) \]

The flexible SVM places different kernels on training samples.