On the design of Kernels

Indo-US joint workshop

Chiranjib Bhattacharyya
Dept of CSA,
Indian Institute of Science

Machine Learning lab
Dept of CSA, IISc
chiru@csa.iisc.ernet.in
http://drona.csa.iisc.ernet.in/~chiru

12-13 Nov 2010

Outline

Motivation

- need to develop ability to learn decision functions on diverse kinds of data
- ability to process large amounts of data

Why kernels are important

- Kernels pave the way for designing learning algorithms which can apply to diverse datasets
- suitable for non-vectorial data e.g. text, songs, protein structures
- Embeds data in a Hilbert space, opening the door for sophisticated algorithms

Introduction to SVMs

Notation

Classifier	$f : \mathscr{X} \rightarrow \{1, -1\}$
Risk	$R(f) = P(f(\mathbf{X}) \neq Y)\ (\mathbf{X}, Y) \sim Q$
$\mathbf{X} \in \mathscr{X},\ \ Y \in \{1, -1\}$	
Training set	$D_m = \{(x_i, y_i) \mid i = 1, \ldots, m\}$
	(x_i, y_i) i.i.d drawn from Q.

- SVMs learn classifiers of the form $f(x) = \text{sign}(\mathbf{w}^\top \Phi(x) + b)$ where $\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \Phi(x_i)$ $\Phi(x)$ is the *feature* map

Introduction to SVMs

Classifier $f(x) = \text{sign}(\sum_{i=1}^{m} y_i \alpha_i k(x, x_i) + b)$

SVM Dual formulation

$$\Gamma(K) = \max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \alpha^\top YKY\alpha \quad (1)$$

$$0 \leq \alpha_i \leq C \qquad \sum_{i=1}^{m} \alpha_i y_i = 0 \quad (2)$$

$\Gamma(K) \Downarrow \implies R(f) \Downarrow$ with high probability.

Introduction to SVMs

$k : \mathscr{X} \times \mathscr{X} \rightarrow \mathbb{R}$
$K_{ij} = k(x_i, x_j)$ $k(x, z) = \Phi(x)^\top \Phi(z)$
Instead of specifying Φ one can specify a kernel function k.

- $k(x, z) = k(z, x)$
- Kernels define a dot product related to Reproducing Kernel Hilbert space.
  For any m $K_{ij} = k(x_i, x_j)$ $1 \leq i, j \leq m$ defines a psd matrix.

On the Problem of Kernel Design

- Learn an optimal Kernel from a library of kernel functions (Lanckriet et al. 2003)
- Learn an optimal kernel from several similarity functions
- Find a *good* classifier when K is uncertain
- All three problems gives rise to extremely interesting optimization problems

Remarks

- For a fixed K computing $\Gamma(K)$ for large m is a Convex QP
- Choice of K maybe guided by the criterion that $\Gamma(K)$ should be low.
- $\Gamma(K)$ is a convex function of K, and could be non-differentiable.

Introduction to Sparse MKL

Multiple Kernel learning(MKL)

$$min_{K \succeq 0} \Gamma(K) \;\; s.t. \; K = \sum_{l=1}^{n} \beta_l K_l, \; \text{trace}(K) = d$$

where K_l are kernel matrices, known aprori.

- K_l is symmetric $m \times m$ psd matrix.
- Semidefiniteness could be automatically assured if $\beta_l \geq 0$

Related work

1. Can be solved as a SDP (Lanckriet et al. 2003),
2. For the case $\beta_l \geq 0$ one can reformulated it as a SOCP(Bach et al. 2004)
3. Can be solved as an iterative algorithm where each iteration requires solving an SVM (Bach 2007, Sonnenberg etal. 2006)
4. Emerging area of research is to investigate alternate formulations for non-sparse MKL (Kloft et al. 2009, Saketha Nath et al. 2009)

Insights into MKL

- MKL defines a *concatenation* of feature maps, $\Phi_l(x)$. $K_l(x,z) = \Phi_l(x)^\top \Phi_l(z)$.

$$K(x,z) = \sum_l \beta_l K_l(x,z) = \Phi(x)^\top \Phi(z)$$

$$\Phi(x)^\top = [\sqrt{\beta_1}\Phi_1(x)^\top, \ldots, \sqrt{\beta_n}\Phi_n(x)^\top]$$

- (Bach et al. 2004) Block-L1 regularization $\left(\sum_{l=1}^{n} \|w_l\|\right)^2$ on the concatenated feature map, recovers the MKL formulation.
- Block L_1 norm leads to sparsity, i.e. drives most of the $\beta_l = 0$
- Often tends to yield the best kernel

Sparsity within groups and non-sparsity between groups

- It promotes sparsity within a group by employing a block-L1 norm and promotes all the groups
- In object categorization often one have the notion of descriptors like shape,color,texture etc.
- (Nilsback and Zisserman 2006) point out that all descriptors must act together.
- However one could have several representations of each descriptor.

MKL in hierarchical feature space

Let each individual example be described by a hierarchical feature map

$$\Phi(x) = [\Phi_1(x)^\top, \ldots, \Phi_n(x)^\top]^\top$$

$$\Phi_j(x) = [\Phi_{j1}(x)^\top, \ldots, \Phi_{jn_j}(x)^\top]^\top$$

$\Phi_{jl}(x)$ is the feature map correspond to the j, lth block.

$$K_{jl}(\mathbf{x}_1, \mathbf{x}_2) = \Phi_{jl}(\mathbf{x}_1)^\top \Phi_{jl}(\mathbf{x}_2)$$

Regularizing hierarchical feature spaces

- Classifier $y = \text{sign}\left(\sum_{j=1}^{n}\sum_{l=1}^{n_j} w_{jk}^{\top}\Phi_{jl}(\mathbf{x}) + b\right)$

A new regularization term

$$\Omega(\mathbf{w}) = \frac{1}{2}\left\{\sum_{j=1}^{n}\left\{\sum_{l=1}^{n_j}\|\mathbf{w}_{jl}\|_2\right\}^{2q}\right\}^{\frac{1}{q}}$$

for $q \geq 1$

- Block L_1 norm is recovered by $q = 1$, and $n = 1$ and $q > 1$ yields non-sparse MKL

Main contributions

Consider the following formulation

$$\min_{\mathbf{w}_{jk},b,\xi_i} \quad \frac{1}{2}\left[\sum_{j=1}^{n}\left(\sum_{l=1}^{n_j}\|\mathbf{w}_{jl}\|_2\right)^{2q}\right]^{\frac{1}{q}} + C\sum_i \xi_i$$

$$\text{s.t.} \quad y_i\left(\sum_{j=1}^{n}\sum_{l=1}^{n_j}\mathbf{w}_{jl}^{\top}\Phi_{jl}(\mathbf{x}_i) - b\right) \geq 1-\xi_i,\ \xi_i \geq 0\ \forall\ i \qquad (3)$$

suitable for hierarchical feature maps.

If $q \to \infty$, the regularization term writes as

$$\frac{1}{2}\max_j\left(\sum_{l=1}^{n_j}\|\mathbf{w}_{jl}\|_2\right)^2$$

Algorithm

We present a convergent algorithm of complexity $O(m^2 n^2 \ln n_t/\varepsilon^2)$, $m =$ no of examples, $n =$ no of groups, $n_t = max_j n_j$ which solves such a problem for arbirary $q \geq 1$.

The primal **Problem (P)** and its dual

$$\min_{\xi_i, b, w_{jk}} \left\{ \max_{\gamma \in \Delta_{n,q^*}} \min_{\lambda_j \in \Delta_{n_j}} \left[\underbrace{\frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n_j} \gamma_j \frac{\| w_{jk} \|^2}{\lambda_{jk}} + C \sum_i \xi_i}_{f(w, \lambda, \gamma)} \right] \right\} \tag{4}$$

The primal **Problem (P)** and its dual

$$\min_{\xi_i, b, w_{jk}} \left\{ \max_{\gamma \in \Delta_{n,q^*}} \min_{\lambda_j \in \Delta_{n_j}} \left[\underbrace{\frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n_j} \gamma_j \frac{\| w_{jk} \|^2}{\lambda_{jk}} + C \sum_i \xi_i}_{f(w, \lambda, \gamma)} \right] \right\} \tag{4}$$

Sion Kakutani Theorem

Let $X, Y \subset \mathbb{R}^d$ be compact and convex. If $g : X \times Y \rightarrow R$ and $g(x, \cdot)$ is continuous and concave in Y, $\forall x \in X$, and if $g(\cdot, y)$ is convex and continuous in $X \ \ \forall y \in Y$, then

$$\min_{x \in X} \max_{y \in Y} g(x, y) = \max_{y \in Y} \min_{x \in X} g(x, y)$$

Dual of **Problem (P)**

$$\min_{\lambda \in \bigotimes \Delta_{n_j}} \max_{\gamma \in \Delta_{n,q^*}} \left\{ \min_{\mathbf{w}} f(\mathbf{w}, \lambda, \gamma) \mid \text{ s.t. } (3) \right\} \tag{5}$$

$\Delta_{n,s} = \{\gamma | \gamma \geq 0, \sum_{i=1}^{n} \gamma_i^s \leq 1\}$ $\Delta_{n,1} = \Delta_n$
Invoking the Lagrangian dual w.r.t w, b, ξ gives

The Dual formulation

$$\min_{\lambda \in \bigotimes \Delta_{n_j}} \max_{\alpha \in A_m, \gamma \in \Delta_{n,q^*}} \sum \alpha_i - \frac{1}{2} \alpha^T \left(\sum_{j=1}^{n} \sum_{k=1}^{n_j} \frac{\lambda_{jk} Q_{jk}}{\gamma_j} \right) \alpha$$

where $A_m = \{\alpha \in \mathbb{R}^m \mid \sum_{i=1}^{m} \alpha_i y_i = 0, 0 \leq \alpha_i \leq C\}$ and
$\left(Q_{jk}\right)_{ih} = y_h y_i K_{jk}(\mathbf{x}_i, \mathbf{x}_h)$ $i, h = 1, \ldots, m$

A first order method based on mirror descent procedure

- The idea is to treat the minimax problem as a minimization problem of the form

$$\min\{G(\lambda_1, \lambda_2, \ldots, \lambda_n) \mid \lambda_j \in \Delta_{n_j},\ j = 1, \ldots, n\}$$

- The function

$$G(\lambda_1, \lambda_2, \ldots, \lambda_n) = \max_{\gamma \in \Delta_n, \alpha \in S_m} \left\{ \mathbf{1}^T \alpha - \frac{1}{2} \alpha^T \sum_{j=1}^{n} \left(\frac{\sum \lambda_{jk} Q_{jk}}{\gamma_j} \right) \alpha \right\} \quad (6)$$

  is convex as it is pointwise maximum of functions which are linear in λ.

- The function could be non-differentiable

Remarks

- For a fixed γ and λ computing optimal α is equivalent to solving an SVM problem with the kernel function $\mathbf{K}_{eff} \equiv \sum_{j=1}^{n} \left(\frac{\sum_{k=1}^{n_j} \lambda_{jk} \mathbf{K}_{jk}}{\gamma_j} \right)$.
- For a fixed α and λ computing optimal γ can be computed in closed form
- The minimization in λ for a fixed α and γ consists of minimizing a convex function over a direct product of simplices

The Algorithm

- Fix λ_k
  - Find α by solving SVM with K_{eff}
  - Find γ_k by a equation
- Compute λ_{k+1} by mirror descent procedure

The $q = \infty$ case was published in NIPS 2009. The general case of $q \geq 1$ is under review with a journal.

Learning kernels from similarity functions

- Often similarity functions are more readily available than kernels.
- Sequence Similarity and Structure Similarity is readily available for Protein structures
- Classifying protein structures is an important problem where both sequence and structure are important

Problem Definition

Given: a set of m training examples with similarity matrices $S_1, S_2, \ldots, S_n \in \mathbb{R}^{m \times m}$, and class labels $\mathbf{y} \in \{+1, -1\}^m$.

Task: learn a kernel matrix K, such that SVM performs well on test data.

Related work

$$\min_{K \succeq 0} \Gamma(K) + \rho \|K - S\|_F^2 . \tag{7}$$

- It is a minimax problem

$$min_K max_\alpha f(\alpha, K)$$

- First studied in (Luss & Aspremont, 2007) and solved using Analytic center cutting plane method.
- (Chen et al et al, 2008) formulated the above problem as Quadratically Constrained Linear Program.
- (Chen et al. 2009) used an alternate loss function $\|K - S\|_F$ which led to Second Order Cone Program.

Related work

- Exploits strong duality

$$min_K max_\alpha f(K, \alpha) = max_\alpha min_K f(K, \alpha)$$

- For Frobenius Norm Loss one could solve $min_K f(K, \alpha)$ in a closed form

$$K^*(\alpha) = (S + \frac{1}{4\rho}(Y\alpha\alpha^\top Y)_+$$

(Luss & Aspremont 2007)

- One can solve the resulting problem

$$max_\alpha f(K^*(\alpha), \alpha)$$

by ACCP procedure

- Will not apply to other losses

Contributions

- Note that $max_\alpha f(K, \alpha)$ is equivalent to solving an SVM. Unlike the state of the art we intend to solve the orginal problem

$$min_K max_\alpha f(K, \alpha)$$

- proposed three formulations which could be solved in an iterative fashion
- Each iteration is no expensive than solving an MKL/SVM
- Leads to scalable solution
- State of the art seems to be specific for the frobenius loss and cannot handle the general case
- Previous work could not handle multiple similarity matrices,

Single Kernel based Formulation

$$\min_{K} \left[\Gamma(K) + \rho \sum_{l=1}^{n} L_l(K - S_l) \right],$$

$$s.t. \qquad K \succeq 0, \;\; \text{trace}(K) = \tau.$$

$L_l(\cdot)$ is a convex, sub-differentiable loss function.

Examples:

1] $L_l(K - S_l) = \sum_i \sum_j |K(i,j) - S_l(i,j)|$

3] $L_l(K - S_l) = \|K - S_l\|_F$

2] $L_l(K - S_l) = \sum_i \sum_j [K(i,j) - S_l(i,j)]^2$

Recall: Multiple Kernel Learning

$$\Gamma(K_1, \ldots, K_n) = \max_{\gamma \in \triangle} \max_{\alpha \in A_m} \left[\alpha^\top \mathbf{1} - \sum_{l=1}^{n} \frac{\alpha^\top Y K_l Y \alpha}{2\gamma_l} \right],$$

$$\Delta_n = \left\{ \gamma \in \mathbb{R}^n \;\middle|\; \gamma \geq 0,\ \gamma^\top \mathbf{1} \leq 1 \right\}.$$

Optimal kernel: $K^* = \sum_{l=1}^{n} \frac{1}{\gamma_l^*} K_l$

Multiple Kernel based Formulation

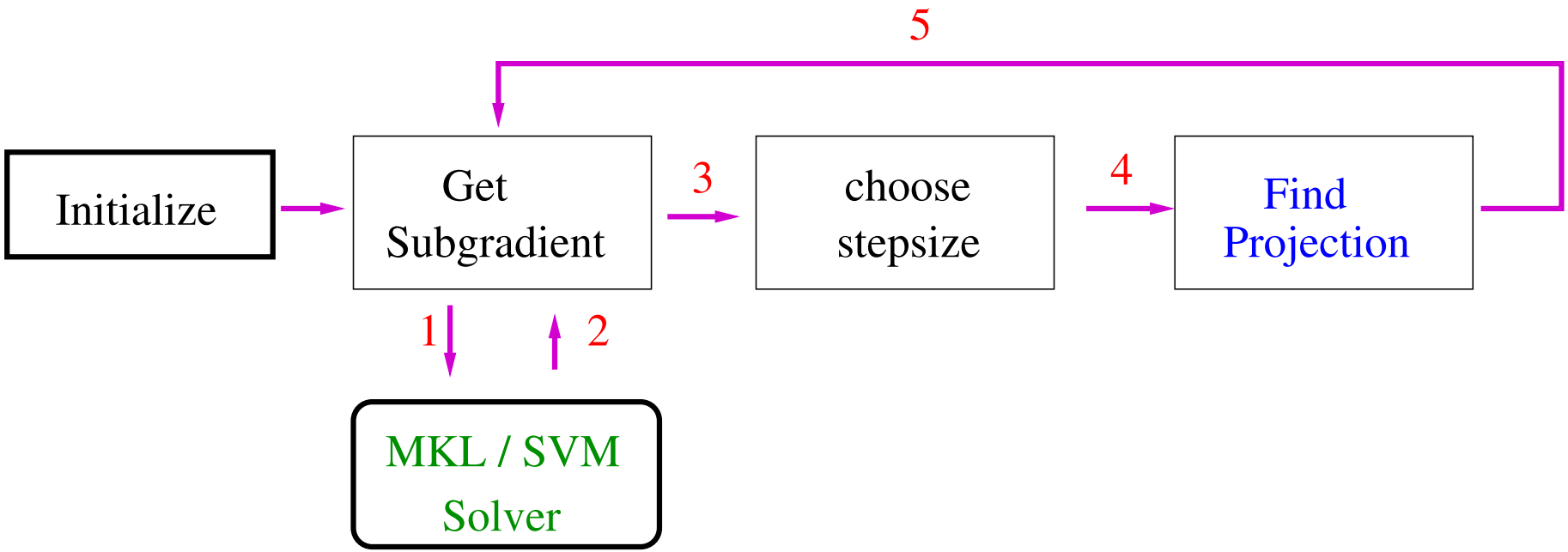
$$\min_{K_1,\ldots,K_n} \left[\Gamma(K_1,\ldots,K_n) + \rho \sum_{l=1}^{n} L_l(K_l - S_l) \right],$$

$$s.t. \;\; K_l \succeq 0, \; \text{trace}(K_l) = \tau, \; l = 1,\ldots,n.$$

Ideas behind the algorithm

- Solve the inner maximization over α for a fixed Kernel or Kernels by solving an SVM or MKL algorithm
- Solve the outer minimization by SubGradient projection.
- We choose Mirror Descent for the Subgardient projection step

Mirror Descent based Algorithm



MD Step 1: Subgradient Computation

- Obtain (α^*, γ^*) by solving the non-sparse MKL.
- Calculate subgradient of Γ as

$$\Gamma'(K_1, \ldots, K_n) = \left(\Gamma'_1, \ldots, \Gamma'_n\right),$$

$$\Gamma'_l := -\frac{1}{2\gamma_l^*} Y\alpha^* \alpha^{*\top} Y.$$

- Compute L'_l - a subgradient of L_l.

MD Step 2: Compute Projection

- Eigen decomposition:

$$\Gamma'_l(K_1^{(t)}, \ldots, K_M^{(t)}) + \rho L'_m(K_m^{(t)} - S_m)$$

$$= \quad V_m^{(t)} \, \mathrm{diag}([d_{1,m}^{(t)} \ldots d_{N,m}^{(t)}]) \; V_m^{(t)\top}.$$

- $\lambda_{i,m}^{(t+1)} := \dfrac{\tau \, \lambda_{i,m}^{(t)} \exp(-\eta_t \, d_{i,m}^{(t)})}{\sum_{j=1}^{N} \lambda_{j,m}^{(t)} \exp(-\eta_t \, d_{j,m}^{(t)})}, i = 1, \ldots, N.$

- $K_m^{(t+1)} := V_m^{(t)} \, \mathrm{diag}([\lambda_{1,m}^{(t+1)} \ldots \lambda_{N,m}^{(t+1)}]) \; V_m^{(t)\top}.$

Algorithmic Convergence

$F^{(t)}$: objective function value at t-th iteration.

F^* : optimal objective value.

$\text{Lip}(F)$: Lipschitz constant of objective function .

Theorem

If the algorithm is initialized with $K_l^{(1)} = \frac{\tau}{m} I, \forall\, n$ *and the stepsizes are chosen as* $\eta_t = \frac{1}{Lip(F)} \sqrt{\frac{2 \log m}{n t}}$

then $\min_{1 \leq t \leq T} F^{(t)} - F^* \leq \tau\, n\, Lip(F) \sqrt{\frac{2 \log m}{T}}$.

Algorithmic Complexity

$O\left(\frac{n^2 \log m}{\varepsilon^2}\right)$ iterations to reach ε-accurate solution.

At every iteration key comutations are:

- Eigen decomposition of n matrices of dimension $m \times m$.
- Solving SVM / MKL with m training examples and n kernels.

Restricted Kernel Learning (RKL)

Eigen value/vector pairs of S_l are: $\{(\mathbf{v}_{i,l}, \lambda_{i,l})\}_{i=1}^{m}$

$$\min_{\mu_1,\ldots,\mu_n \in \mathbb{R}^N} \left[\Gamma(K_1, \ldots K_n) + \rho \sum_{l=1}^{n} \ell_m(\mu_l - \lambda_l)\right],$$
$$s.t. \quad K_l = \sum_{i=1}^{m} \mu_{i,l} \mathbf{v}_{i,l} \mathbf{v}_{i,l}^{\top}, \quad \mu_{i,l} \geq 0, \quad \sum_{i=1}^{l} \mu_{i,l} = \tau.$$

$\ell_m : \mathbb{R}^N \rightarrow \mathbb{R}$ is a convex, subdifferentiable loss function.

MD Algorithm for RKL

- Obtain (α^*, γ^*) by solving the non-sparse MKL with $K_l = \sum_i \mu_{i,l}^{(t)} \mathbf{v}_{i,l} \mathbf{v}_{i,l}^\top, \forall l$.
- Compute a subgradient of ℓ_l: $\ell_l'(\mu_l^{(t)}) = [\, \ell_{1,l}'(\mu_m^{(t)}) \ \ldots \ \ell_{m,l}'(\mu_l^{(t)}) \,]^\top$.
- $g'^{(t)}_{i,l} := -\frac{1}{2\gamma_l^*} \alpha^{*\top} Y \mathbf{v}_{i,l} \mathbf{v}_{i,l}^\top Y \alpha^* + \ell_{i,l}'(\mu_l^{(t)})$.
- $\mu_{i,l}^{(t+1)} := \dfrac{\tau \mu_{i,l}^{(t)} \exp\left(-\eta_t g'^{(t)}_{i,l}\right)}{\sum_{j=1}^m \mu_{j,l}^{(t)} \exp\left(-\eta_t g'^{(t)}_{j,l}\right)}, \ i = 1, \ldots, m.$

Learning classifiers from Uncertain Data

- State of the art Protein structure classification algorithms use Kernels.
- Kernels on protein structures, $K(P_1, P_2)$, are available where a Protein structure is represented as a point cloud $P = \{(\mathbf{x}_i \in \mathbb{R}^3; i = 1, \ldots, n_i\}$
- However $\mathbf{x}_i$ is often noisy
- In Protein Data Bank, a resolution information is often available which can be understood as a measure of uncertainty

Problem Statement

Consider the problem of designing classifiers when the Kernel matrix is noisy.

$$K_{ij} = \overline{K}_{ij} + Z_{ij}$$

Z_{ij} is the uncertainty, $\overline{\mathbf{K}}$ is a valid kernel.

The SVM dual

$$\begin{aligned} max_{\alpha \in A_m, t} \quad & \alpha^\top e - \frac{1}{2} t \\ \text{s.t.} \quad & \alpha^\top YKY\alpha \leq t \end{aligned}$$

Chance Constrained Program (CCP)

$$\begin{aligned} max_{\alpha \in A_m, t} \quad & \alpha^\top e - \frac{1}{2} t \\ \text{s.t.} \quad & Prob(\alpha^\top Y(\overline{\mathbf{K}} + Z) Y \alpha \leq t) \geq 1 - \varepsilon \end{aligned}$$

This is often NP hard for arbitrary uncertainty.
Challenge is to design tractable approximations.

Chance Constrained Program (CCP)

$$\begin{aligned} max_{\alpha \in A_m, t} \quad & \alpha^\top e - \frac{1}{2} t \\ \text{s.t.} \quad & Prob(Tr((\overline{\mathbf{K}} + Z) Y \alpha \alpha^\top Y) \leq t) \geq 1 - \varepsilon \end{aligned}$$

Nature of Uncertainty

- Z_{ij} are **independent**
- $E[Z_{ij}] = 0$
- Distribution
  1. **Gaussian** distribuion with known variance σ_{ij}^2
  2. Unknown distribution with $\mathbf{a_{ij}} \leq \mathbf{Z_{ij}} \leq \mathbf{b_{ij}}$

The main result

Final result

$$min_{t,\alpha \in A_m} \quad \frac{1}{2}t - \sum_i \alpha_i$$

$$\text{s.t.} \quad \alpha^\top Y \overline{\mathbf{K}} Y \alpha + \kappa \sqrt{\sum_{ij} \beta_{ij} \alpha_i^2 \alpha_j^2} \leq t$$

★ Gaussian ($RSVM^{(g)}$)
- $\kappa = -\Phi^{-1}(\varepsilon)$
- $\beta_{ij} = \sigma_{ij}^2$

★ FiniteSupport ($RSVM$)
- $\kappa = \sqrt{2\log(1/\varepsilon)}$
- $\beta_{ij} = f(a_{ij}, b_{ij})$

$$Prob(Tr(ZV) \geq u) \leq exp\left\{ -\frac{1}{2} \frac{u^2}{\|\beta' * V\|_F^2} \right\}$$

where, $\beta_{ij} = f(a_{ij}, b_{ij})$ and $\beta'_{ij} = \beta_{ij}^{\frac{1}{2}}$

Contributions

Key Contributions

Solved CCP involving matrix valued uncertainty, as an SOCP

- **Gaussian** uncertainty
- **Finite support** but arbitrary uncertainty
- Derived a **large deviation inequality**
- **Specialized** solver
- i.i.d uncertainty $\longrightarrow$ **SVM**
- General case is non-convex but yields a convex solution in special cases
- Incorporates resolution information in Real world problem of **protein structure classification**

Presented in ICML 2010

Acknowledgements

This is joint work with
Prof. A. BenTal, Dr. J. Saketha Nath, Prof. K. R. Ramakrishnan, Dr Sourangshu Bhattacharya
Students: Dinesh, Raman, Sahely, Achintya, Vikram
Supported by: Yahoo! Faculty award, IBM Faculty award