Johnson-Lindenstrauss, Concentration and applications to Support Vector Machines and Kernels

Devdatt Dubhashi

Department of Computer Science and Engineering,
Chalmers University,
dubhashi@chalmers.se

Functional Inequalities in Discrete Spaces with Applications
Simons Institute, Sept. 30–Oct. 4 2013.

Outline I

Outline I

Statement

Johnson–Lindenstrauss Flattening Lemma '84

Given $0 < \varepsilon \leq 1/2$ and $x_1, \cdots x_n \in R^D$, there exists a map $f : R^D \to R^k$ with $k = O\left(\frac{1}{\varepsilon^2} \log n\right)$ such that for all $i, j \in [n]$,

$$(1-\varepsilon)\|x_i - x_j\|_2^2 \leq \|f(x_i) - f(x_j)\|_2^2 \leq (1+\varepsilon)\|x_i - x_j\|_2^2$$

- Note that the mapping is into a dimension that is independent of the original dimension and only depends on the number of points.
- For every n, there exists a set of n points requiring target dimension $k = \Omega(\frac{1}{\varepsilon^2} \log n / \log(1/\varepsilon))$ (Alon, 2003).

JL vis Measure Concentration on Sphere

Original proof:

- Random linear map is orthogonal projection on k dim subspace.
- Then use measure concentration on sphere (Dvoretzky's Theorem).
- Still maybe most intuitive proof, but today elementary proofs (Indyk–Motwani, Achlioptas, Gupta and Gupta, Matousek, Naor ...)

Applications of Johnson–Lindenstrauss

Proximity Problems Computing nearest neighbours in high dimensions (for example nearest documents to web queries) can be done in much lower dimensions.

Online Problems Answering queries on–line

Data Streaming/Storage Large data which cannot be in RAM arrives sequentially and we can only store "sketches".

Compressed Sensing One pixel camera ...

Machine Learning Classification and clustering of high dimensional data.

Probabilistic Method

Distributional JL

Given any $0 < \varepsilon \leq 1/2, \delta > 0$ there is a random linear mapping $A : R^D \to R^k$ with $k = O\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}\right)$ such that for any unit vector $x \in R^D$,

$$\mathbf{P}\left[(1-\varepsilon) \leq \|A(x)\|_2^2 \leq (1+\varepsilon)\right] \geq 1 - \delta$$

- Note that the mapping is universal and projected dimension depends only on ε and δ, not on original dimension D.
- Lower bound of $k = \Omega(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$ (Jayram–Woodruff, Kane–Meka–Nelson 2011).

Distributional implies Deterministic

Take $\delta = 1/n^2$ so $k = O\left(\frac{1}{\varepsilon^2}\log n\right)$ and then for each pair $i, j \in [n]$,

$$\mathbf{P}\left[(1-\varepsilon)\|x_i - x_j\|_2^2 \leq \|A(x_i) - A(x_j)\|_2^2 \leq (1+\varepsilon)\|x_i - x_j\|_2^2\right] \geq 1 - 1/n^2$$

Hence by a simple union bound, the same statement holds for all $\binom{n}{2}$ pairs $i, j \in [n]$ simultaneously with probability at least $1/2$. By a classic application of the probabilistic method this implies the existence of a determinstic linear mapping which is an approximate isometry.

Explicit Mapping

Ultra Simple Linear Map

$$A(x) := \frac{1}{\sqrt{k}} Mx$$

where the entries of the $k \times D$ matrix M are

i.i.d. Gaussian

$$M_{i,j} \sim N(0,1)$$

Outline I

Input data as Feature vectors

Input in many machine learning problems is in the form of very high dimensional vectors, often sparse:

- Documents as bag of words: x_i is no. of occurences of word i in a document
- Network traffic $x_{i,j}$ is no. of packets sent from i to j
- User ratings x_i is rating for movie i by a user on Netflix
- Streaming Making an update requires computing a sparse matrix vector product.

The classification problem setup

Training Data Let P be a (unknown) distribution and

$$D_n = \{(\mathbf{x}_i, y_i) |\ i.i.d\ (\mathbf{x}_i, y_i) \sim P, i = 1, \ldots, n\}$$

Here $\mathbf{x}_i \in R^D$ and $y_i = \pm 1$

Classifier A function $f : R^D \to \{-1, +1\}$ constructed using the data D_n as training set.

Test Data Let $(\mathbf{x}, y) \sim P$. Then want to minimize the probability of misclassification

$$R(f) = P(f(\mathbf{x}) \neq y)$$

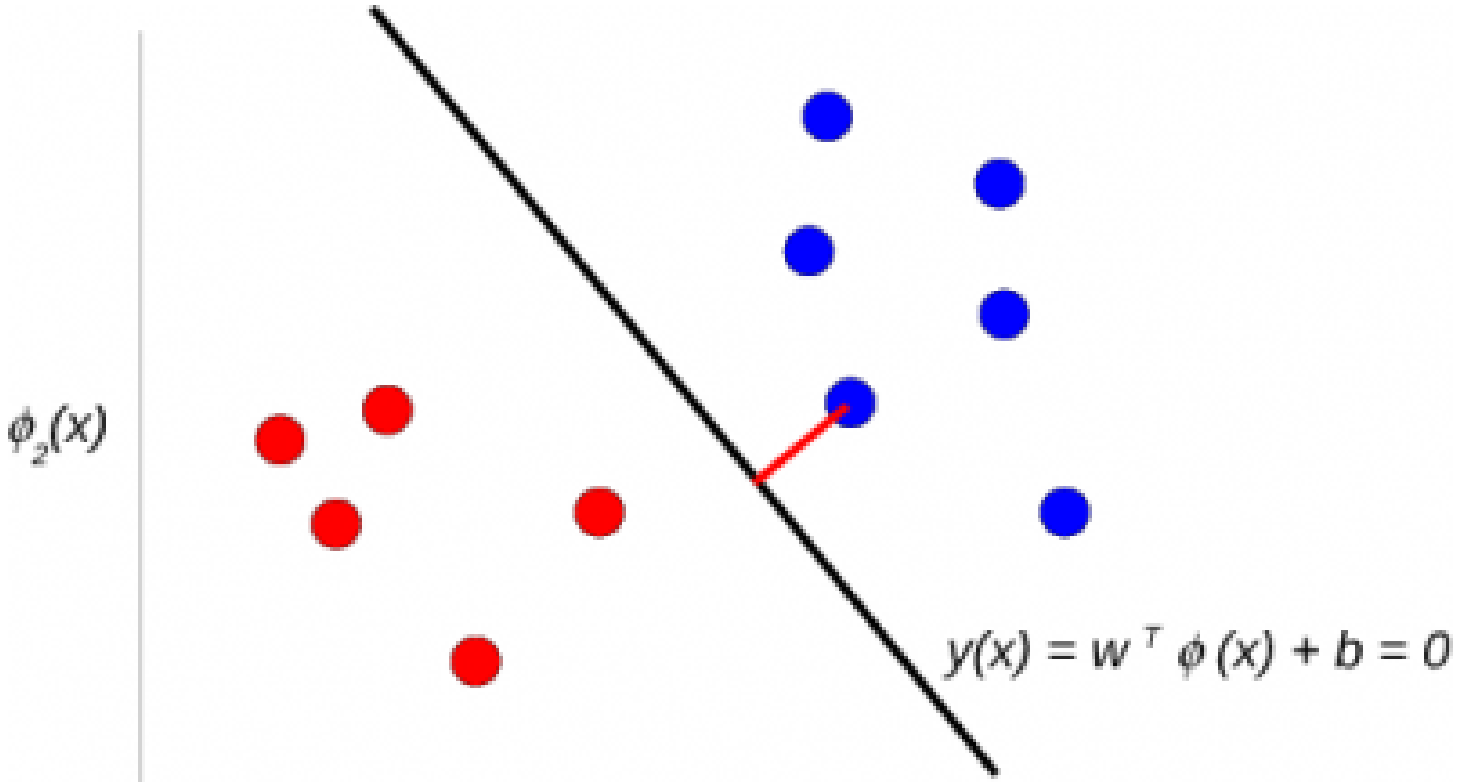
Linear classifiers

Linear classifiers

$$y = sign(w^\top x + b)$$

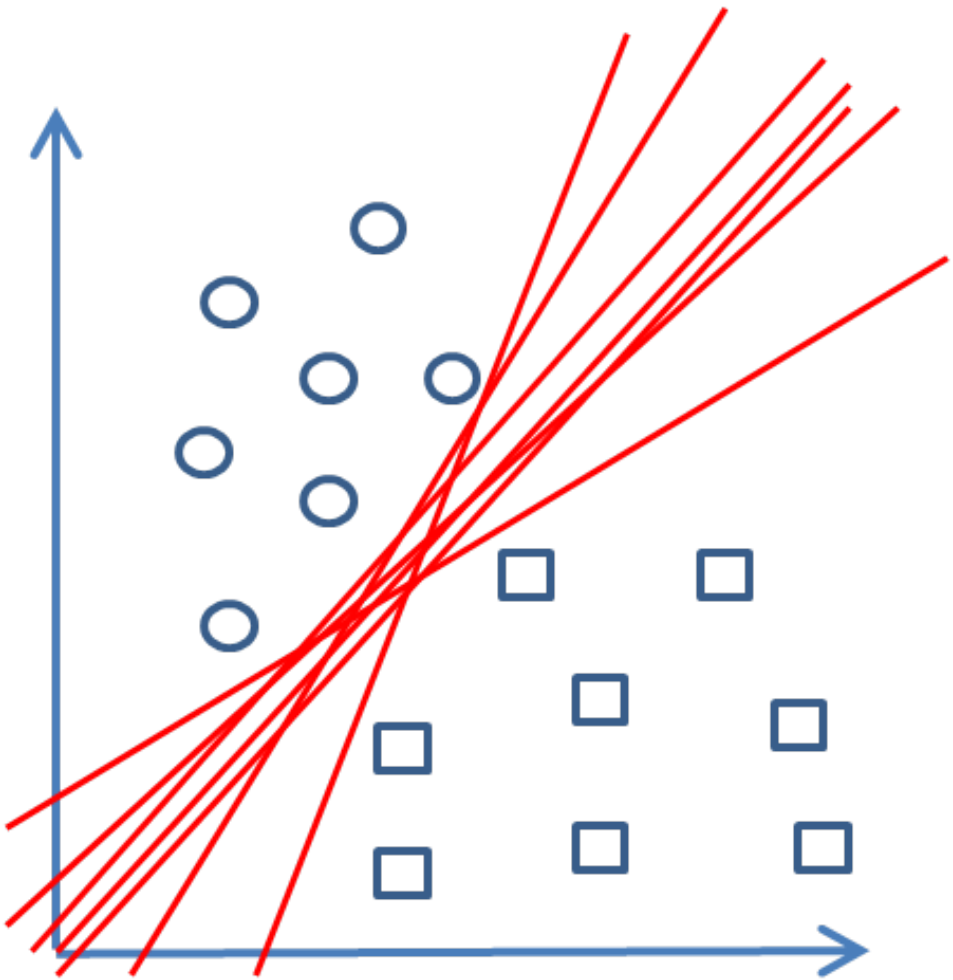
require satisfying

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \; i = 1, \ldots, m$$



Good classifiers: Margins

Can be many linear classifiers:

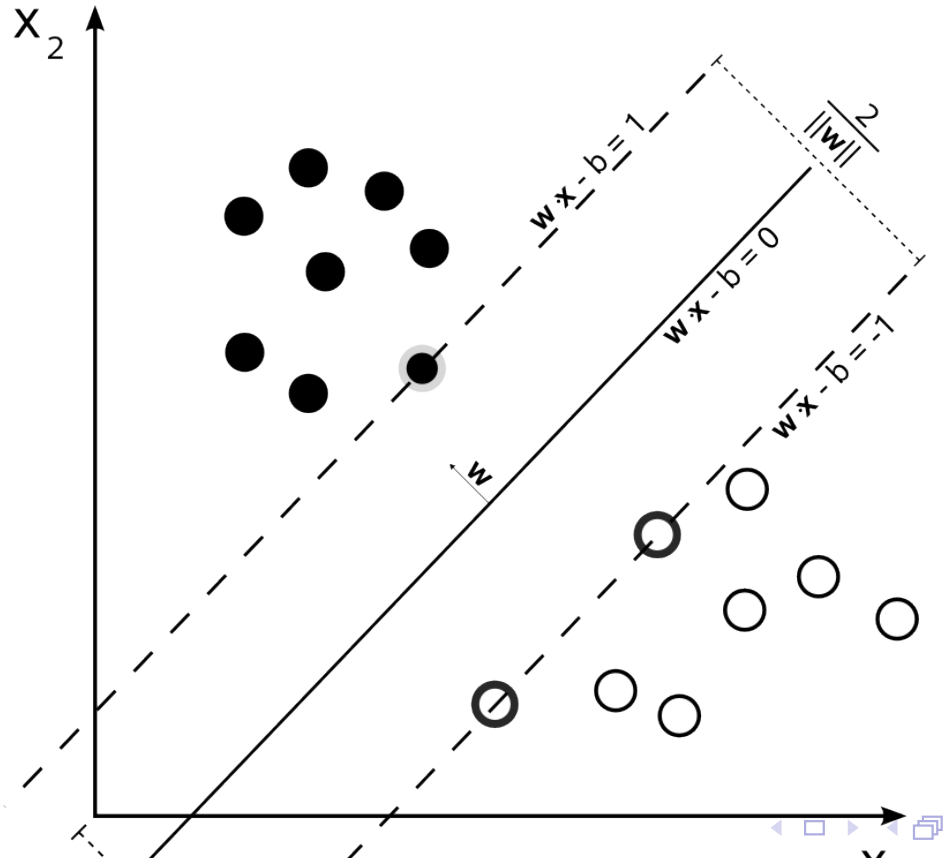


Max Margin

One can work with parallel hyperplanes

$$\mathbf{w}^\top \mathbf{x} + b = 1, \;\; \mathbf{w}^\top \mathbf{x} + b = -1$$

with the classifier fixed at $\mathbf{w}^\top \mathbf{x} + b = 0$. Maximize the margin, $2/\|\mathbf{w}\|$, the distance betwen the hyperplanes!



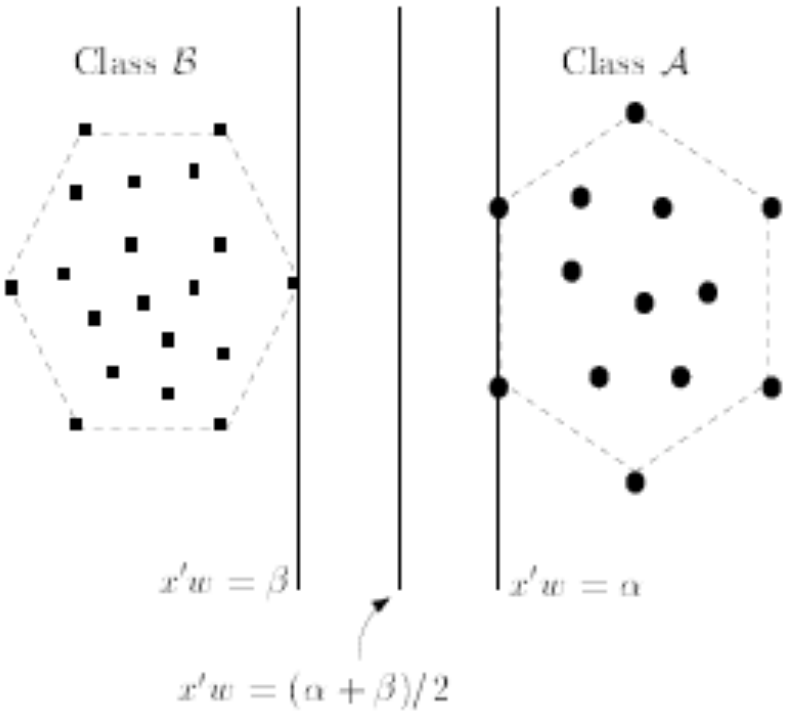
The SVM formulation

Classifier

$$f(x) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$$

$$\begin{aligned} \text{minimize}_{\mathbf{w},b} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 && (1) \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \; i = 1, \ldots, m && (2) \end{aligned}$$

SVMs represent data as Convex hulls



$$\text{minimize} \|z_1 - z_2\|^2$$

$$z_1 \in Conv\left(\{\mathbf{x}_i | y_i = 1\}\right)$$

$$z_2 \in Conv\left(\{\mathbf{x}_i | y_i = -1\}\right)$$

- SVMs computes the distance between Convex hulls
- Will exploit this later

The Dual formulation

$$\text{maximize}_\alpha \; -\frac{1}{2}\sum_{ij}\alpha_i\alpha_j y_i y_j x_i^\top x_j + \sum_{i=1}^{m}\alpha_i$$

$$\text{subject to } \; 0 \leq \alpha_i, \sum_i \alpha_i y_i = 0$$

KKT conditions

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \implies \mathbf{w} = \sum_i \alpha_i y_i x_i$$

$$\frac{\partial L}{\partial b} = 0 \implies \sum_{i=1}^{m} \alpha_i y_i = 0$$

$$\alpha_i(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1) = 0, \; \alpha_i \geq 0$$

Support vectors and separating hyperplane

- Support vectors correspond to $\{i|\alpha_i > 0\}$.
- Consequence of KKT conditions

$$\alpha_i > 0 \implies y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$$

- The separating hyperplane is given as

$$\sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + w_0 = 0,$$

or

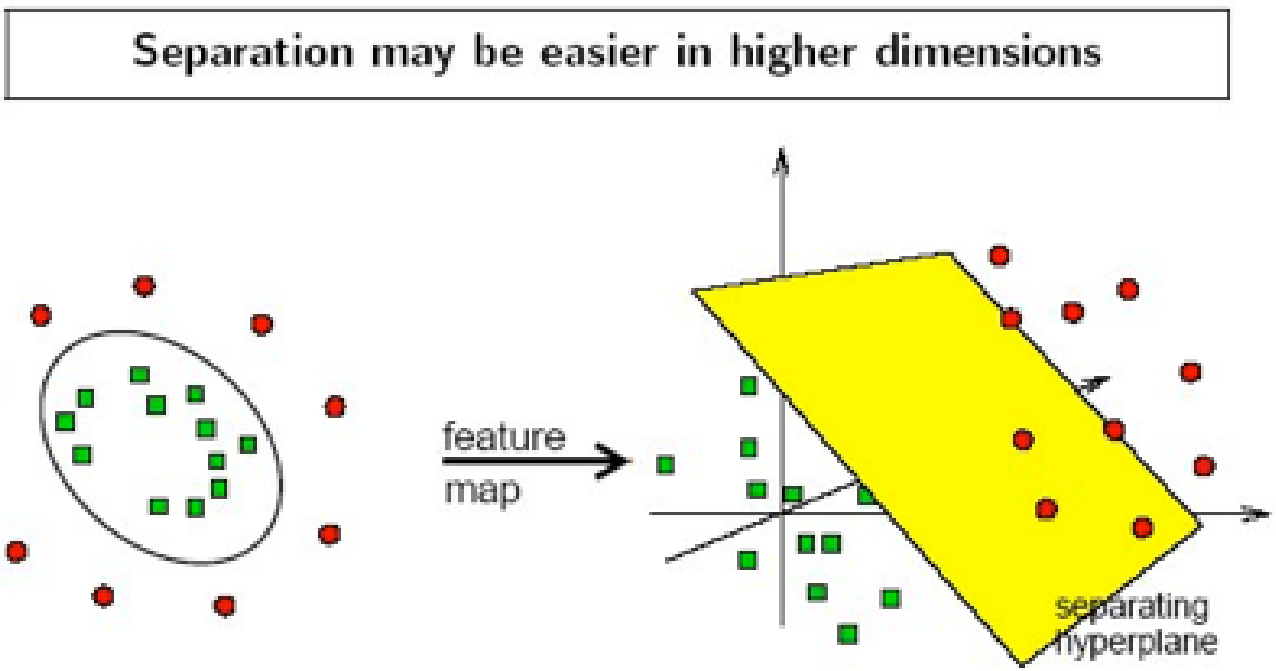
$$\sum_{\alpha_i > 0} \alpha_i y_i K(x_i, x) + w_0 = 0.$$

That is, be computed only in terms of the dot products or the kernel.

SVMs as non-linear classifiers

- Sometimes data may not be separable by a linear classifier
- ... but may be linearly separable after embedding in higher dimensions.
- Example in 2 dimensions (x_1, x_2) that is not linearly separable but becomes separable after embedding:

$$\mathbf{x} = (x_1, x_2) \mapsto \Phi(\mathbf{x}) := (x_1^2, \sqrt{2}x_1x_2, x_2^2).$$



The Kernel Trick

- Note that we only need $K_{i,j} := \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$ for our SVM solver.
- Thus one can implicitly define the embedding via the matrix K.

Mercer's Condition

The matrix K is defined by an embedding into a Hilbert space iff K is positive semi–definite.

Popular Kernels

Linear $K(\mathbf{x},\mathbf{z}) := \mathbf{x}^T\mathbf{z}$ corresponding to Φ being identity.

Quadratic $K(\mathbf{x},\mathbf{z}) := (\mathbf{x}^T\mathbf{z})^2$ or $(1+\mathbf{x}^T\mathbf{z})^2$.

Polynomial $K(\mathbf{x},\mathbf{z}) := (\mathbf{x}^T\mathbf{z})^d$ or $(1+\mathbf{x}^T\mathbf{z})^d$.

Radial Basis Functions $K(\mathbf{x},\mathbf{z}) := \exp\left(-\gamma\|\mathbf{x}-\mathbf{z}\|_2^2\right)$ correspomds to an infinite dimensional embedding.

Often data may be presented only via a similarity kernel in the form of a psd matrix with embedding implicit and which may be high or infinite dimensional. hence important to be able to work only with kernel representation.

Running Time of SVM Solvers

SVM-light	$O(n^2 d)$ (dual QP)
Lib-Lin	$O(nd \log(1/\rho)$ (dual coordinate descent).
Pegasos	$O(d/\lambda\rho)$ (primal, stochastic gradient/subgradient)

Parameters:

- n number of data points in training set.
- d dimension
- ρ optimization tolerance
- λ regularization parameter

JL and SVMs: Take 1

Approximate Max Margin

If a set of n points in R^D is separable with margin γ, then the projected set of points to R^k with $k = O(\frac{1}{\varepsilon^2} \log \frac{n}{\delta})$ is separable with margin at least $(1-\varepsilon)\gamma$ with probability at least $1-\delta$.

- One can obtain a near–optimal classifier in the original space by solving an SVM in the projected space.
- However need to account for computation time for projection: naively takes time $O(kD)$ per data point.
- Recall that input data is high dimensional i.e. large D, but is often sparse $s := \|x\|_0 << D$ non–zero coordinates in any input point.

JL Sparser and Faster

Achlioptas '01 $M_{i,j} = 0$ with prob. 1/3.

Ailon–Chazele '06 $x \mapsto PHDx$ (P is sparse random matrix, H is Hadamard transform and D is diagonal ± 1. $O(D \log D + k^3)$.

Ailon–Liberty '08 $O(d \log k + k^2)$ also using Hadamard transform.

Ailon–Liberty, Krahmer-Ward '11 $O(D \log D)$ but sub–optimal $k = O(\frac{1}{\varepsilon^2} \log n \log^4 D)$.(Uses [Rudelson–Vershynin '08], Dudley inequality etc.)

Kane–Nelson 2010

Dimension $k = O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$

Sparsity each column of matrix has at most $s = O(\frac{1}{\varepsilon} \log \frac{1}{\delta})$ non–zero entries.

Computing time $O(s\|x\|_0)$.

Concentration for Faster JL

- Challenge is that the hashing based construction of Kane and Nelson (2010) involves dependent random variables.
- How can we salvage concentration when there is dependence?
- Kane and Nelson (2010) use an explicit and elaborate combinatorial computation of higher moments.
- Can we prove similar results with a unified machinery?

Negative Dependence and Concentration

- The hashing based construction of Kane and Nelson involves random variables with a natural negative dependence property.
- Many classical concentration results are valid under various negative dependence properties e.g. negative association:

$$E[f(X_i, i \in I)g(X_j, j \in J] \leq E[f(X_i, i \in I)]E[g(X_j.j \in J)]$$

for disjoint index sets I, J and non–decresing functions f, g.

Negative Association in Kane-Nelson

The coordinates of the projected vector $V_1, \cdots, V_k$ for an input point $x \in R^D$ with non–negative coordinates in the Kane–Nelson hashing construction are negatively associated, and each coordinate is sub–Gaussian, hence the same concentration results apply as in the i.i.d. case.

Outline I

Data as Kernels

- However, if data is only given in the form of similarity kernels, then the associated mapping Φ may be very high dimensional or even infinite ...
- Instead an approach via LP type abstract optimization problems.

Random sampling for LP

- Take a random sample of constraints (much smaller than full set).
- Solve LP problem on random sample of constraints and retain basis that determines optimum.
- If some constraint outside sample is violated, update basis.
- Seidel's linear time algorithm for LP in constant dimension.

Abstract LP type problems

- Smallest enclosing ball of a set of points
- Smallest enclosing ellipsoid of aset of points
- Smallest enclosing annulus of a set of points.
- Distance between two polyhedra (SVM!)

What is common?

Small basis Optimal solution determined by a small number δ of input objects (points, linear inequalities ..) independent of total number of input objects.

Monotonicity Solution value only increases (or decreases) as you add more input objects

Locality If solution increases (or decreases) by adding more input objects, it does so already by adding to the at δ objects in the basis.

LP Type problem: abstract framework

A pair (H, w)

- H finite set of constraints (or points)
- $w : 2^H \to R \cup \{-\infty, \infty\}$ objective function.

For any $F \subseteq G \subseteq H$

Monotonicity $w(F) \leq w(G)$

Locality if $w(F) = w(G) \neq -\infty$,

$$w(G \cup \{h\}) > w(G) \quad \to \quad w(F \cup \{h\}) > w(F)$$

Interpretation: $w(G)$ is minimum value subject to constraints in G

Basis and Combinatorial Dimension

$(H.w)$ LP-type problem.

Basis A basis of $F \subseteq H$ is a minimal subset $B \subseteq F$ such that $w(B) = w(F)$.

Combinatorial Dimension Size of largest basis, Δ

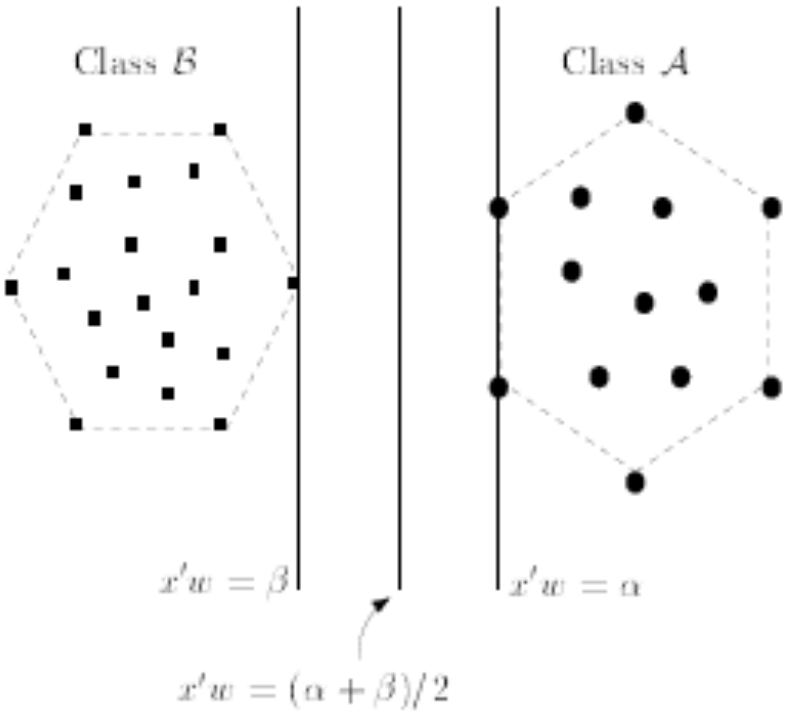
Examples: Combinatorial dimension

Smallest enclosisng ball $D+1$.

Linear prog $D+1$.

Distance between hyperplanes $D+2$

SVMs represent data as Convex hulls



$$\text{minimize} \|z_1 - z_2\|^2$$

$$z_1 \in Conv\left(\{\mathbf{x}_i | y_i = 1\}\right)$$

$$z_2 \in Conv\left(\{\mathbf{x}_i | y_i = -1\}\right)$$

- Finding max margin hyperplane is finding distance between two convex polytopes.

Solving LP type problems

Two primitives needed (B is a basis and h a constraint)

Violation test $w(B \cup h) > w(B)$?

Basis update $B' \leftarrow \text{basis}(B \cup h)$.

Random Sampling Algorithm for LP Type

Input: (H, w), Output: A basis B of H so $w^* = w(B)$
$S =$ random subset of H of size Δ
$B =$ basis$(\emptyset, S)$, $V =$violators$(G - S, B)$
While $(|V| > 0)$
$R =$ choose a random subset of size $\Delta - |B|$ from V
$B =$ basis(B, R)
$V =$ violators$(G - R, B)$
end while
return B

Sub–Exponential Algorithm for LP-type

Matousek, Sharir, Welzl

The expected running time of the algorithm **lp-type** is $O(n \cdot 2^{\sqrt{\Delta \log \Delta}})$.

- Linear time for LP in constant dimension.
- What about SVM?

JL for SVMs: Take 2

Input: D(Dataset), Output: SV, set of Support vectors
$S =$ random subset of D of size Δ
$SV =$ svmsolver({ },S), $V =$ KKT violators of $D - S$
While $(|V| > 0)$
$R =$ choose a random subset of size $r - |SV|$ from V
$SV' =$ svmsolver(SV,R), $SV = SV'$
$V =$ KKT violators from non-sampled dataset
end while
return SV

Running time for SVM

- For dimension D data, the combinatorial dimension for SVMs is $D+2$ and running time is $O(n \cdot 2^{\sqrt{D \log D}})$.
- .... but if we're willing to accept near–optimal margin we can work in much lower dimension: $k = (\frac{1}{\varepsilon^2} \log n)$ and still retain margin at least $(1-\varepsilon)\gamma^*$
- ... and running time becomes $O(n \cdot 2^{\sqrt{\log n \log\log n}}) = o(n^2)$.
- Moreover we do not need to explicitly project: All computations can be done with data presented only in kernel form – very important for SVM methods!

Outline I

Back to the Primal

- Number of support vectors can grow linearly with size of training set
- Makes kernel methods with dual formulation expensive for large scale problems
- Many recent advances in first order methods for the primal leading to extremely fast algorithms that scale to very large problems.
- But for a given kernel, how can we get hold of corresponding embedding – may even be infnite dimensional!

Random sampling!

Mercer's Theorem

Any positive definite kernel can be expanded as

$$k(x,y) = \int_z p(z)\lambda_z < \phi_z^*(x), \phi_z(y) > dz$$

- Randomly sample i with prob. proportional to λ_i
- Use approximation

$$\hat{k}(x,y) = \frac{\int_z p(z)dz}{n} \sum_{i=1}^{n} < \phi_{z_i}^*(x), \phi_{z_i}(y) >$$

Invarant kernels

- Kernels invariant under action of a symmetric group.
- Fourier basis gives infinite expansion.

Gaussian RBF kernel

- $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{\sigma^2}\right)$.
- Basis functions $\phi_z(x) = e^{i<z,x>}$
- $k(x, y) = E_z[< \phi_z^*(x), \phi_z(y) >]$

Random Kitchen Sinks for Gaussian RBF

Rahimi–Recht 2007

Let $Z \in R^{n \times d}$ with

- $n = O(\frac{1}{\varepsilon^2} d \log \frac{1}{\delta})$
- $Z_{i,j}$ i.i.d $\mathcal{N}(0, \sigma^2)$.

Then with the approximation with basis functions $\phi_j(x) = \frac{1}{\sqrt{n}} e^{i(Zx)_j}$

$$\hat{k}(x,y) = \frac{1}{n} \sum_{j=1}^{n} e^{i(Zx_j - Zy_j)}$$

$$P\left(|k(x,y) - \hat{k}(x,y)| > \varepsilon\right) < \delta.$$

Follows by measure concentration for Lipschitz functions with Gaussian measure.

JL for SVMs, Take 3: Faster approximation

- Le, Sarlos and Smola (2013) give a faster approximation of the kernel using a different matrix which is a product of Hadamard, binary and permutation matrices.
- Reduces multiplication time from nD to $n \log D$.
- For a special case they prove a concentration result using an adaptation of Gaussian concentration of Lipschitz functions.
- Applies to other kernels that depend on $\|\mathbf{x} - \mathbf{y}\|$
- Conjecture: same is true with Kane–Nelson construction reducing multiplication time to $s\|x\|_0$.

Open Problems

- Develop methods for concentration with negative dependence.
- Can faster JL constructions (Kane and Nelson) be used to approximate kernels?


V. Jethava, C. Bhattacharyya, R. Hariharan
"A Randomized Algorithm for Large Scale Support Vector Learning",
NIPS 2007


Q. Le, T. Sarlos, and A. Smola
"Fastfood - Computing Hilbert Space Expansions in loglinear time"
J. Machine Learning Res. W&CP 28 (3) 244–252, 2013


A. Rahimi and B. Recht,
"Random Features for Large Scale Kernel Machines"
NIPS 2007