CS:E4830 Kernel Methods in Machine Learning

Lecture 11 : Large-scale Learning - Linear and Kernel Methods

Rohit Babbar

27th March, 2019

Today's topics

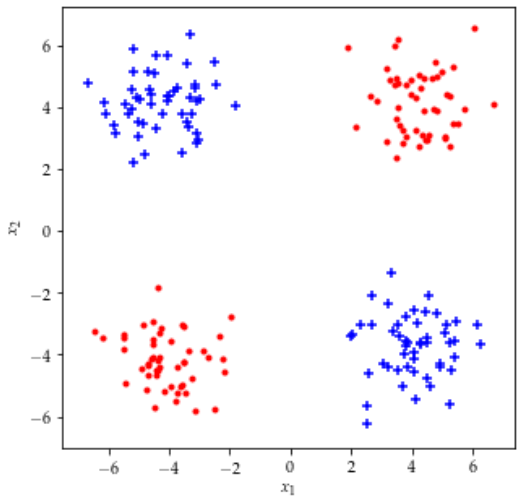
Two methods for **large-scale learning** :

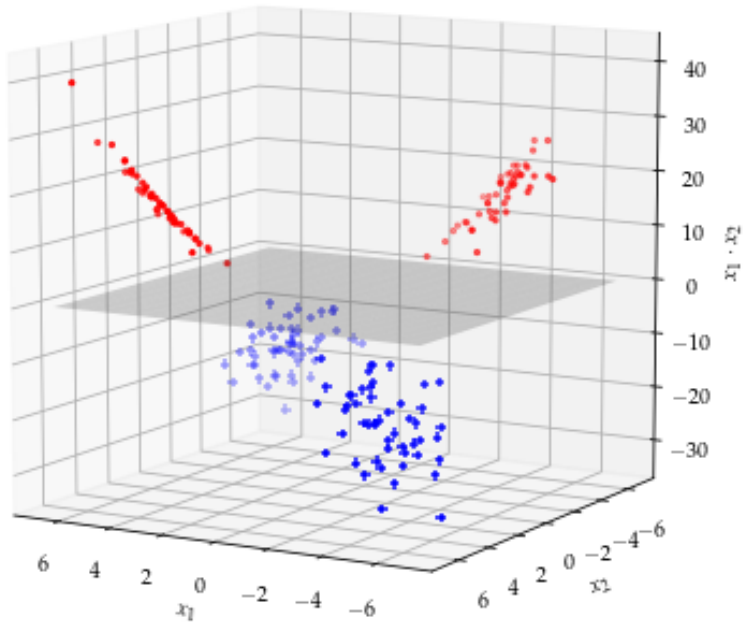
- Large-scale Linear classfication for high-dimensional data (taking co-ordinate descent as an optimization algorithm)
- Non-linear classification (Bochner Theorem for Kernels)

Recall - Picture from Lecture 1

Motivation for using Kernels

- **Implicit mapping of data** to high-dimensional space where the data is separable
- The kernel matrix K gives the pair wise computation required for wide range of training algorithms





Kernel Matrix Bottleneck

In the modern day big data setup where it is not difficult to get millions (or even bigger) of training samples :

- Computing a million $\times$ million kernel matrix is not trivial
  - For instance, one needs to invert the kernel matrix for kernel ridge regression - complexity $O(N^3)$ for N training data points
- The computational bottle-neck also limits hyper-parameter tuning

Large-scale Linear Classification - 1st part of today's lecture

- In some cases, we may have **high dimensional input features** such as classification problems involving **textual data** :
  - Classification of data as in Wikipedia, News Articles, Research articles
  - Ranking of queries/Documents in Searching Engines
  - Recommendation engines such as Amazon, e-bay (recall from a previous lecture)
  - Web-advertising

In these cases, due to *linear separability of the data*, it might just be enough to use linear classifiers, i.e., no feature mapping required

- $f(x) = \mathbf{w}^T\mathbf{x}$ is the decision function, i.e. $\phi(\mathbf{x}) = \mathbf{x}$ (aka linear kernel)

Large-scale Linear Classification - 1st part of today's lecture

- In some cases, we may have **high dimensional input features** such as classification problems involving **textual data** :
  - Classification of data as in Wikipedia, News Articles, Research articles
  - Ranking of queries/Documents in Searching Engines
  - Recommendation engines such as Amazon, e-bay (recall from a previous lecture)
  - Web-advertising

In these cases, due to *linear separability of the data*, it might just be enough to use linear classifiers, i.e., no feature mapping required

- $f(x) = \mathbf{w}^T\mathbf{x}$ is the decision function, i.e. $\phi(\mathbf{x}) = \mathbf{x}$ (aka linear kernel)
- How we might check linear separability of the given dataset in high dimensions ?

Linear Classification on Non-linear Feature Mappings - 2nd part of today's lecture

Another approach is the following :

1. Sometimes, we can also project data to high dimensions by some **explicit** feature map
2. Then, train a linear method on the data in higher dimension

- We also saw this approach when we discussed LibShortText for classification of short textual data where bigrams were used and then a linear classifier was used.
- Today, we will see another example of this approach for translation invariant kernels such as the Gaussian kernel by invoking a property of positive definite function using Bochner's Theorem.

Large-scale linear classification

High Dimensional Data - Text Classification Example from eBay

```
Tickets calgary flames vancouver canucks 2 tickets sept 24 2012 row 2
Stamps  stamps italy kingdom used high value f 9608
Music   brahms j hungarian dances cd new
Jewelry & Watches       5 colors optional hotaru fashion silicone date calendar jelly candy sport watch
Tickets free legoland ticket for friend w legoland pass member expires 8 31 12
Books   specimen book of monotype printing types vol 2 c 1971 garamond to othello
Tickets 4 new york jets vs new england patriots tickets 11 22 sec 336 row 17 10 yd
Art     lowe 1858 antique fern botanical print polypodium lycopodioides
Art     m c escher black white print house of stairs
Books   days in the lives of social workers 54 professionals tell real life stories
Music   bizet g carmen sung in english cd new
Stamps  parliament house cent fdc perth fdi pmk 2
Jewelry & Watches       faceted multigem sterling silver bracelet by silverrushstyle
Tickets michael jackson the immortal tour 8 15 12 staples 2 lower level
Baby    nuk giant bottle nipple adult baby directly from germany
Travel  pacsafe metrosafe 100 anti theft shoulder bag chocolate
Stamps  post office 1984 175 th anniv fdc perth museum fdi pmk
Books   prentice hall geometry textbook
Stamps  timor 50 th anniv 1992 pse corel sea townsville frank
Stamps  life saving clubs 75 th anniv fdc henley beach fdi pmk
Jewelry & Watches       25 mm jade faceted drop giada gemstone loose beads
Tickets seattle seahawks vs tennessee titans 2 tickets sec 107 seats 5 6 aug 11 at 7 pm
Travel  olympia rolling shopper toteblack 20 x 5 x 4 tn
Jewelry & Watches       10 k white gold round diamond lucky horseshoe fashion ring wg size 7 unique gift
```

- Each row is training example in the form of LABEL SPACE INPUT-VECTOR
- Consider 100,000 s such examples

Term Frequency and Inverse Document Frequency

TfIDF based data representation for sparse text data

- Bag of Words
  - Converts **sequence of words** representation of a document to a **set of words**
- Tf-IDF weighting of every term in the document
  - Term-Frequency $\times$ Inverse Document Frequency
  - Term-Frequency (log normalized) of term t in document d

$$tf(t, d) = \log(1 + f_{t,d})$$

where $f_{t,d}$ is the number of times t appears in document d

  - Inverse Document Frequency - How frequent is the term across documents

$$idf(t) = \log(1 + \frac{N}{n_t})$$

where n_t is the number of times term t appears from the training set of N documents

Learning Objective for Linear Classification

Learning process typically involves learning a decision boundary by minimizing an objective function:

- Given training set $\mathbf{x}_i, y_i$ such that $y_i = \pm 1$ for binary classification
- Template of the optimization problem being solved

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{N} Loss(y_i, f(\mathbf{x}_i)) + \lambda \; Reg(\mathbf{w})$$

  where $f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i$ under the linear model setup
- $Loss(\mathbf{w}; \mathbf{x})$ represents the empirical error which is data dependent
- $Reg(\mathbf{w})$ is the regularization term to avoid complex models or to prevent over-fitting

L1-regularized Linear SVM

For this lecture, we will consider L1-regularized with Squared Hinge Loss

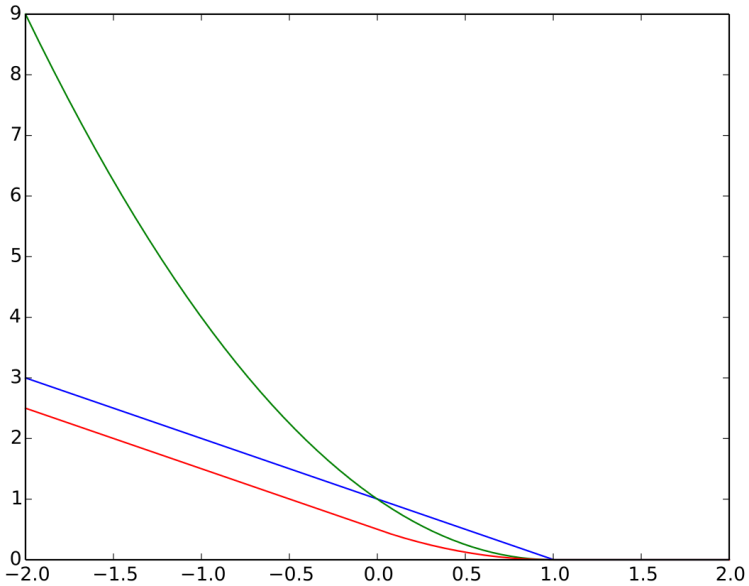
$$f(\mathbf{w}) = \min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||_1 + \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2$$

L1-regularized Linear SVM

For this lecture, we will consider L1-regularized with Squared Hinge Loss

$$f(\mathbf{w}) = \min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||_1 + \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2$$

Sqaured hinge loss $(max(0, 1 - z))^2$ in green and normal hinge loss $max(0, 1 - z)$ in blue

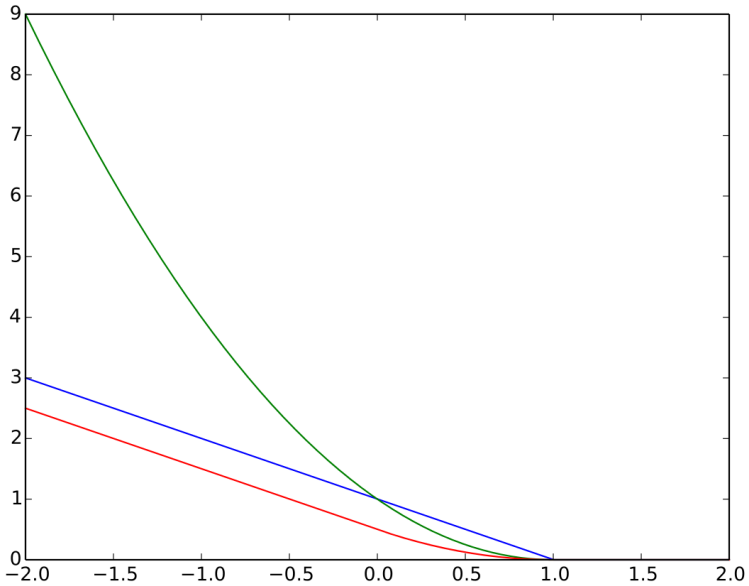


L1-regularized Linear SVM

For this lecture, we will consider L1-regularized with Squared Hinge Loss

$$f(\mathbf{w}) = \min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||_1 + \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2$$

Sqaured hinge loss $(max(0, 1 - z))^2$ in green and normal hinge loss $max(0, 1 - z)$ in blue



- How does $||\mathbf{w}||_1$ look like, in 1D, and 2D?

L1-regularized linear SVM

For this lecture, we will consider L1-regularized with Squared Hinge Loss

$$f(\mathbf{w}) = \min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||_1 + \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2$$

L1-regularized linear SVM

For this lecture, we will consider L1-regularized with Squared Hinge Loss

$$f(\mathbf{w}) = \min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||_1 + \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2$$

To which of the following, the above objective is likely to correspond to ?

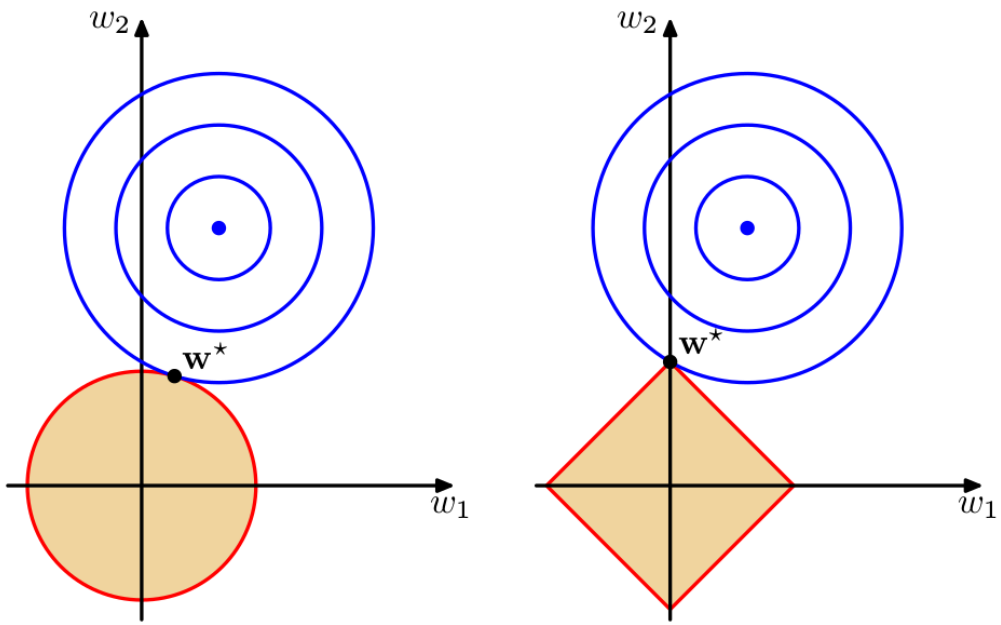


Figure: Figure from Bishop's book

L1-regularized linear SVM

For this lecture, we will consider L1-regularized with Squared Hinge Loss

$$f(\mathbf{w}) = \min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||_1 + \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2$$

To which of the following, the above objective is likely to correspond to ?

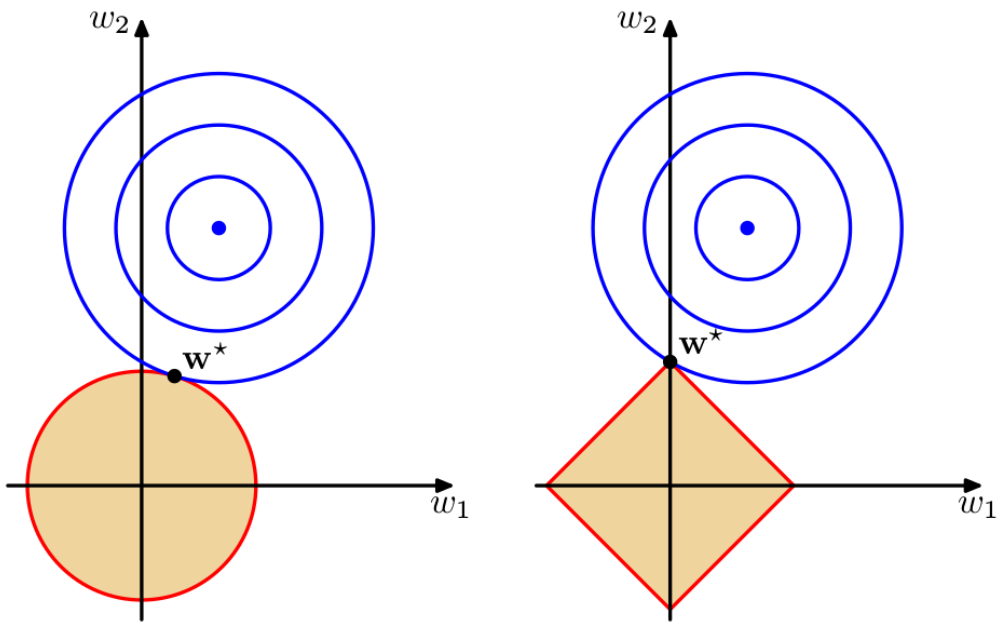


Figure: Figure from Bishop's book

- How about sparsity inducing property?

How to minimize the Objective?

L1-regularized with Squared Hinge Loss function - $\mathbb{R}^D \mapsto \mathbb{R}$

$$f(\mathbf{w}) = \min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||_1 + \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2$$

How to minimize the Objective?

L1-regularized with Squared Hinge Loss function - $\mathbb{R}^D \mapsto \mathbb{R}$

$$f(\mathbf{w}) = \min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||_1 + \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2$$

How to minimize it?

Can we do the following ?

$$\nabla f(\mathbf{w}) := \nabla_w \{\min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||_1 + \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2\} = \mathbf{0}$$

L1-regularized SVM

- **L1-regularized with Squared Hinge Loss function** - $\mathbb{R}^D \mapsto \mathbb{R}$

$$f(\mathbf{w}) = \min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||_1 + \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2$$

- The first term in the above objective function is the regularizer

$$\frac{\lambda}{2}||\mathbf{w}||_1 = \frac{\lambda}{2}\sum_{j=1}^{D}|w_j| \text{ i.e., the sum of absolute values over features}$$

L1-regularized SVM

- **L1-regularized with Squared Hinge Loss function** - $\mathbb{R}^D \mapsto \mathbb{R}$

$$f(\mathbf{w}) = \min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||_1 + \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2$$

- The first term in the above objective function is the regularizer

$$\frac{\lambda}{2}||\mathbf{w}||_1 = \frac{\lambda}{2}\sum_{j=1}^{D}|w_j| \text{ i.e., the sum of absolute values over features}$$

  - Is it smooth (differentiable)?
- The second is the squared hinge loss function

$$L(\mathbf{w}) = \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2 \text{ i.e., the sum over training samples}$$

L1-regularized SVM

- **L1-regularized with Squared Hinge Loss function** - $\mathbb{R}^D \mapsto \mathbb{R}$

$$f(\mathbf{w}) = \min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||_1 + \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2$$

- The first term in the above objective function is the regularizer

$$\frac{\lambda}{2}||\mathbf{w}||_1 = \frac{\lambda}{2}\sum_{j=1}^{D}|w_j| \text{ i.e., the sum of absolute values over features}$$

  - Is it smooth (differentiable)?
- The second is the squared hinge loss function

$$L(\mathbf{w}) = \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2 \text{ i.e., the sum over training samples}$$

  - Is it smooth (differentiable)?

Co-ordinate Descent Scheme

- We will minimize the objective $f(\mathbf{w})$ using Co-ordinate Descent (Figure source : Wikipedia)

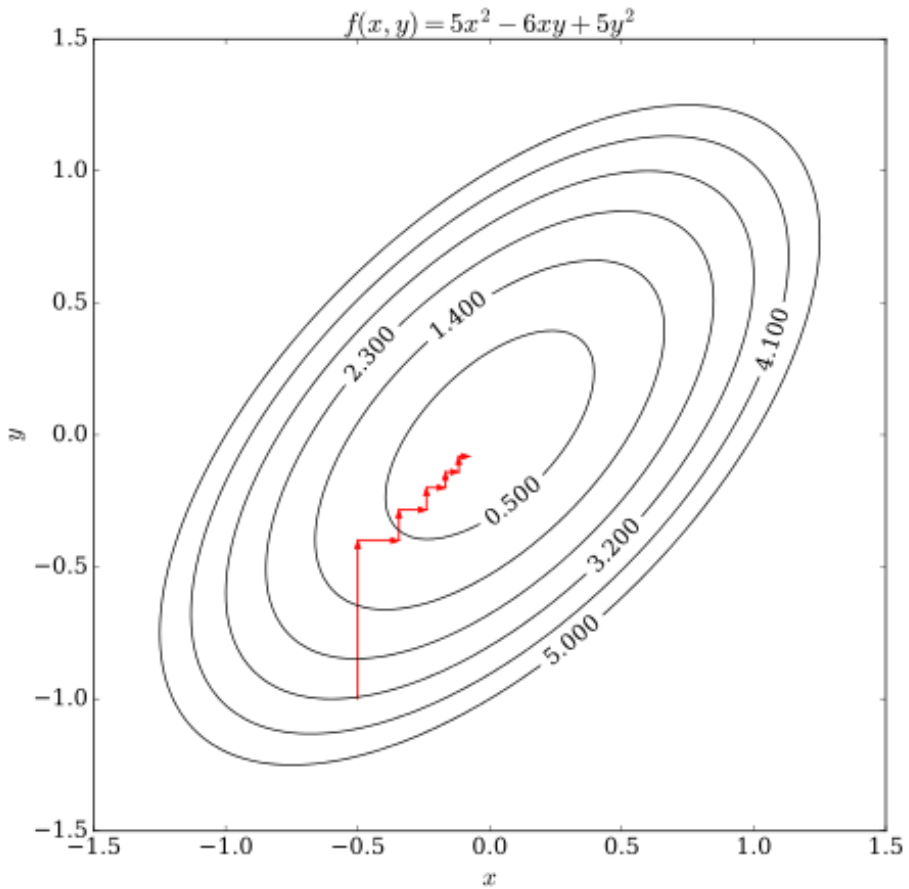
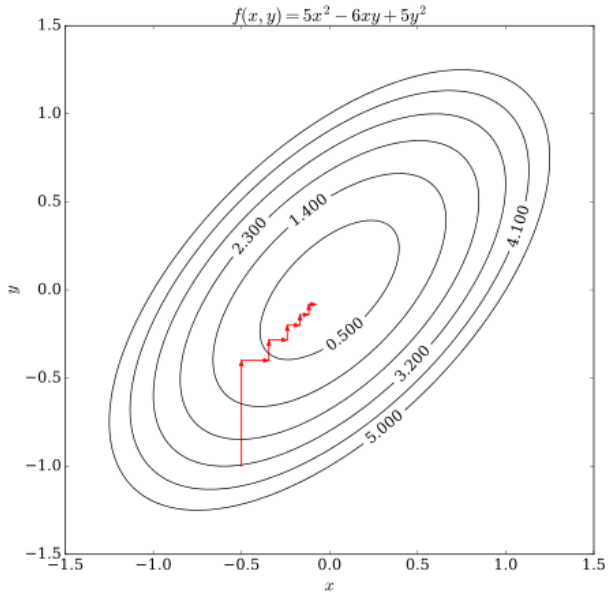


Figure: co-ordinate decent on a function (depicted by level curves) of two variables

Co-ordinate Descent Scheme

- Co-ordinate Descent



- Algorithm

1: Start with some initialization of $\mathbf{w}^0$
2: **for** $\{k = 0;$ until convergence $; k++\}$ **do**
3: Choose a co-ordinate j from $d = 1 \ldots D$
4: Choose step size α
5: Update $\mathbf{w}_j^k$ to $\mathbf{w}_j^{k+1} := \mathbf{w}_j^k - \alpha \nabla_{\mathbf{w}_j} f(\mathbf{w})$
6: **end for**

Co-ordinate Descent - Convex and Differentiable Functions

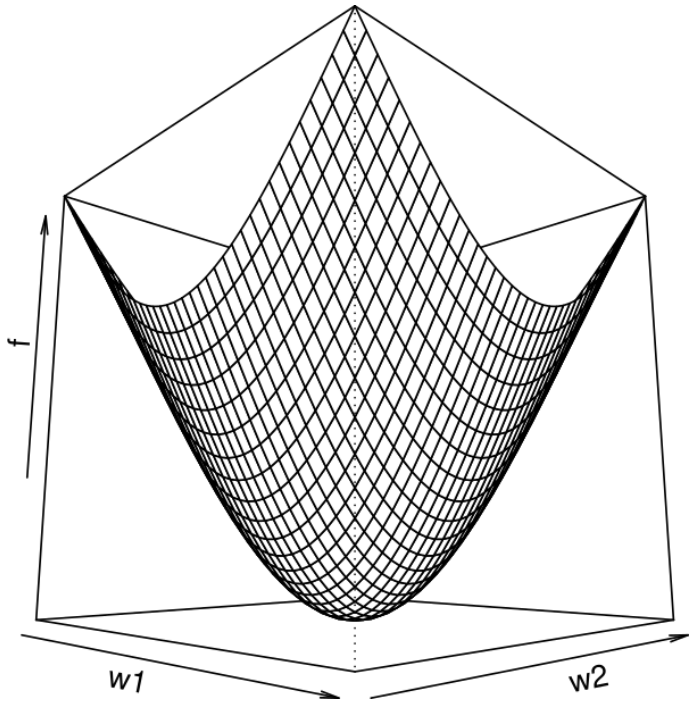
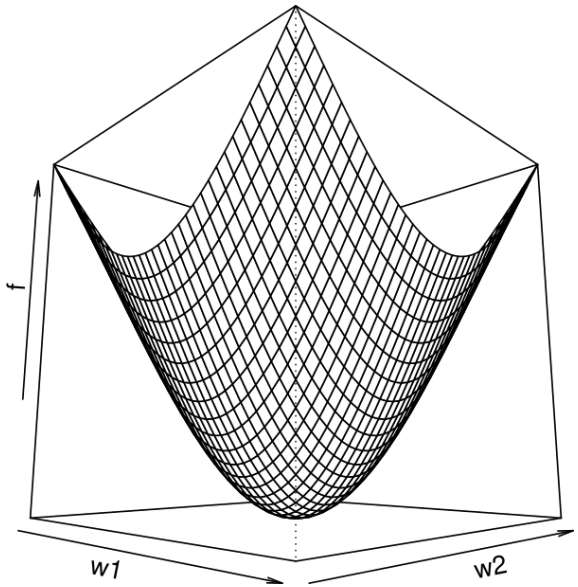


Figure: Based on Lecture notes of Ryan Tibshirani, CMU
`https://www.cs.cmu.edu/~ggordon/10725-F12/slides/25-coord-desc.pdf`

Given a convex and differentiable function $f : \mathbb{R}^D \mapsto \mathbb{R}$, if we are at a point $\mathbf{w}$ such that $f(\mathbf{w})$ is minimized along each co-ordinate axis. Then, does it mean that we have found a global minimizer?
In other words, is the following true?

$$f(\mathbf{w} + z.\mathbf{e}_j) \geq f(\mathbf{w}) \text{ for all } z, \text{ and } j = 1, \ldots, D \implies f(\mathbf{w}) = min_{\mathbf{u}} f(\mathbf{u})$$

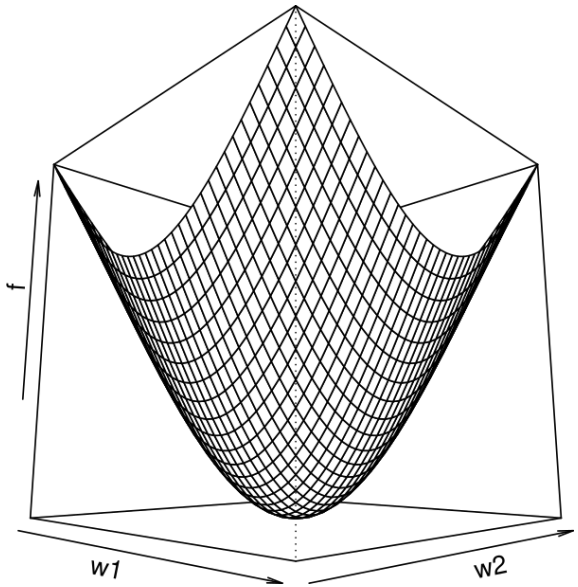
Co-ordinate Descent - Convex and Differentiable Functions



How about the gradient ?

$$\nabla_{\mathbf{w}} f(\mathbf{w}) = \left[\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}_1}, \frac{\partial f(\mathbf{w})}{\partial \mathbf{w}_2}, \ldots, \frac{\partial f(\mathbf{w})}{\partial \mathbf{w}_D} \right]$$

Co-ordinate Descent - Convex and Differentiable Functions



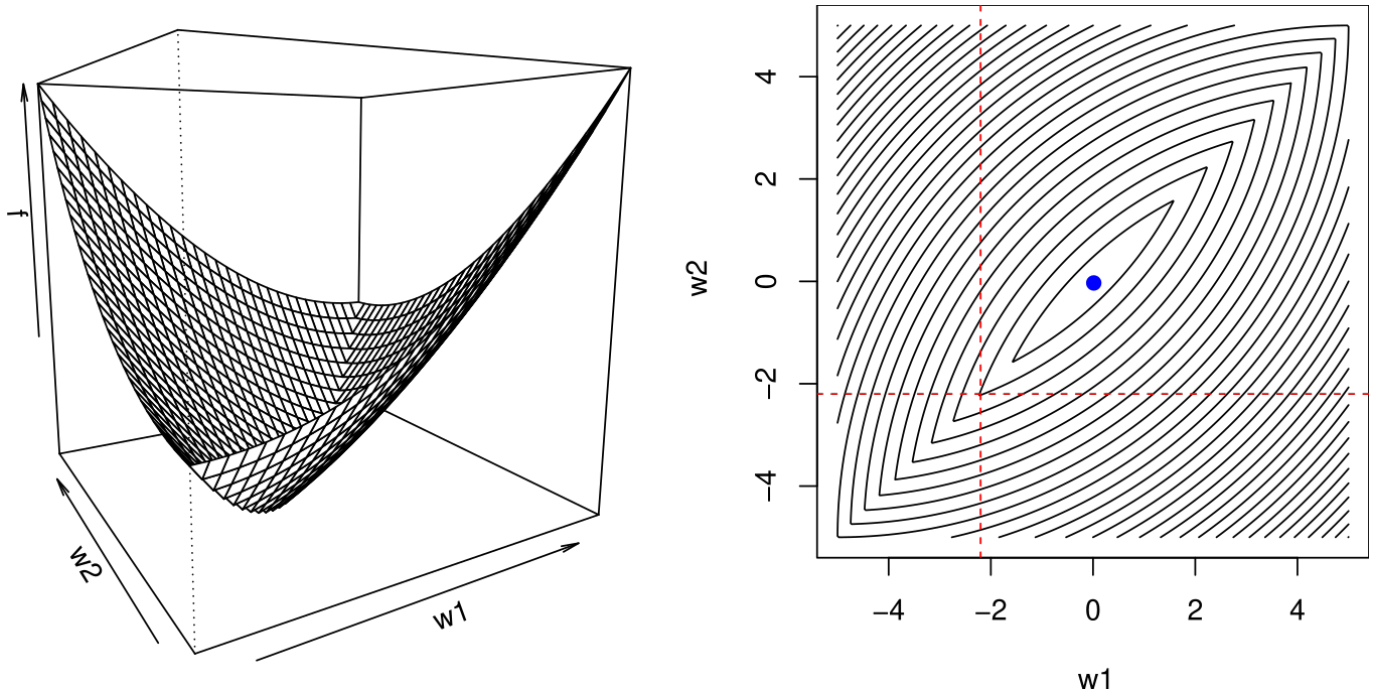
How about the gradient ?

$$\nabla_{\mathbf{w}} f(\mathbf{w}) = \left[\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}_1}, \frac{\partial f(\mathbf{w})}{\partial \mathbf{w}_2}, \ldots, \frac{\partial f(\mathbf{w})}{\partial \mathbf{w}_D} \right]$$

Recall the 1st order condition for convexity. For all $\mathbf{w}$ and $\mathbf{w}'$ (not necessarily on the co-ordinate axes)

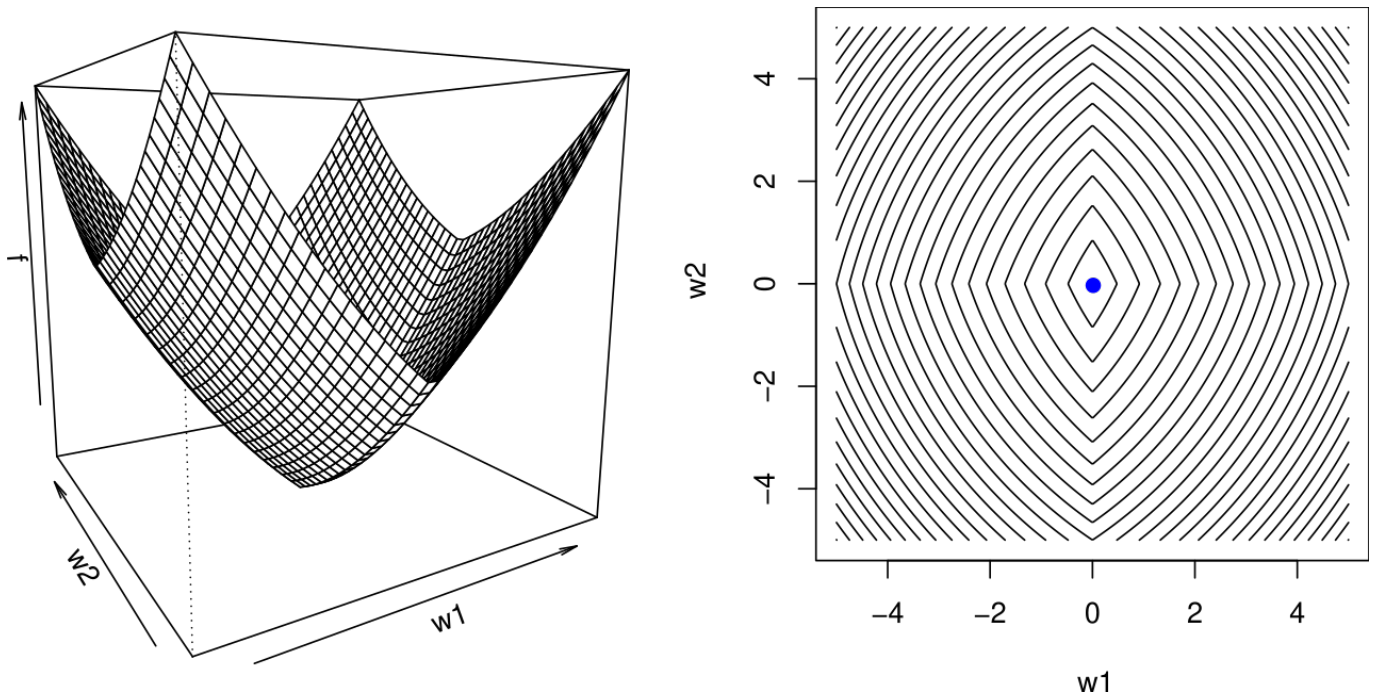
$$f(\mathbf{w}') \geq f(\mathbf{w}) + \nabla_w f(\mathbf{w})^T (\mathbf{w}' - \mathbf{w})$$

Co-ordinate Descent - Convex and Non-smooth Functions



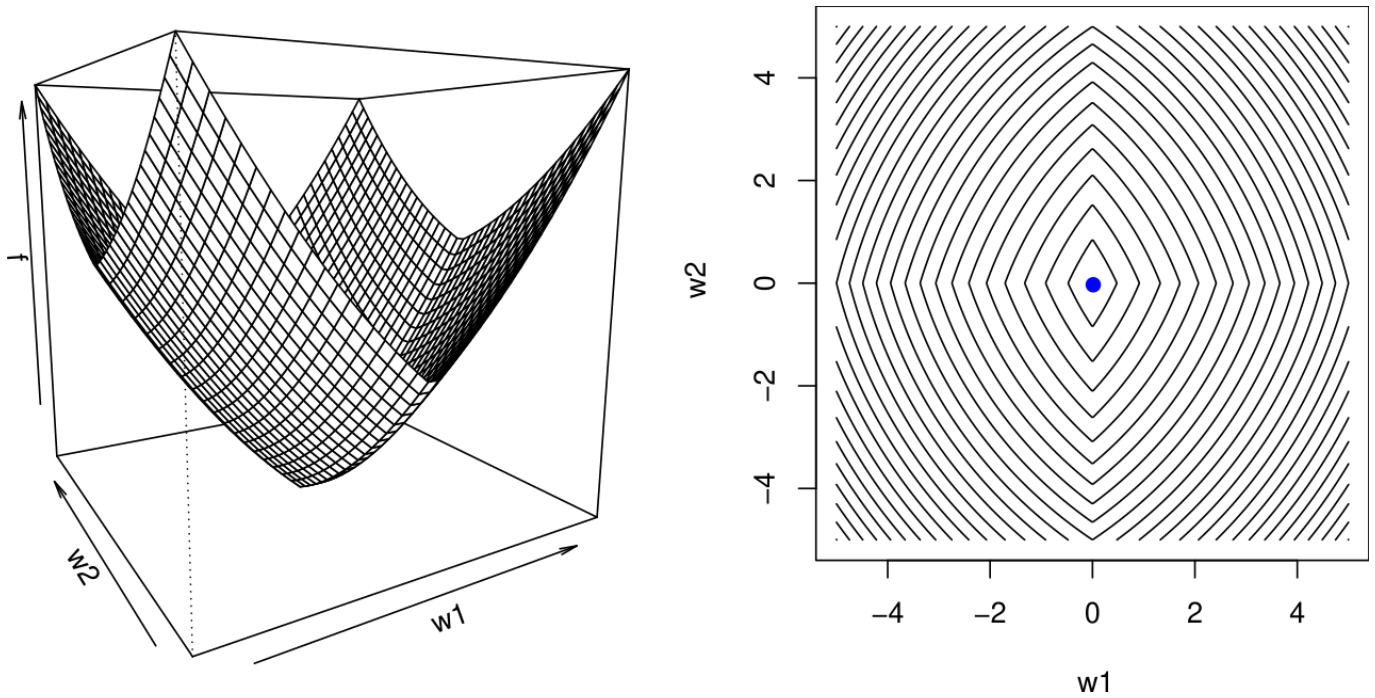
- How about non-smooth functions, such as above?

Co-ordinate Descent Scheme - Separable Non-smooth part



How about when the non-smooth part is separable over co-ordinates(directions)? That is, $f(\mathbf{w}) = g(\mathbf{w}) + \sum_{j=1}^{D} h_j(\mathbf{w}_j)$ where $g(.)$ and $h(.)$ represent the smooth and non-smooth parts of the objective function.

Co-ordinate Descent Scheme - Separable Non-smooth part



$$f(\mathbf{w}') - f(\mathbf{w}) \geq \nabla_{\mathbf{w}} g(\mathbf{w})^T (\mathbf{w}' - \mathbf{w}) + \sum_{j=1}^{D} \left[h_j(\mathbf{w}'_j) - h_j(\mathbf{w}_j) \right]$$

$$= \sum_{j=1}^{D} \left[\underbrace{\nabla_{\mathbf{w}_j} g(\mathbf{w})(\mathbf{w}'_j - \mathbf{w}_j) + h_j(\mathbf{w}'_j) - h_j(\mathbf{w}_j)}_{\geq 0} \right] \geq 0$$

Recall the Objective Function

- **L1-regularized with Squared Hinge Loss function** - $\mathbb{R}^D \mapsto \mathbb{R}$

$$f(\mathbf{w}) = \min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||_1 + \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2$$

Recall the Objective Function

- **L1-regularized with Squared Hinge Loss function** - $\mathbb{R}^D \mapsto \mathbb{R}$

$$f(\mathbf{w}) = \min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||_1 + \sum_{i=1}^{N}(\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i))^2$$

- Let $I(\mathbf{w})$ and $b_i(\mathbf{w})$ respectively be the indices and coresponding penalty for the points with non-zero training error:

$$b_i(\mathbf{w}) := 1 - y_i \mathbf{w}^T \mathbf{x}_i \text{ and } I(\mathbf{w}) := \{i | b_i(\mathbf{w}) > 0\}, i = 1 \ldots N$$

Co-ordinate Descent Method [1]

- For co-ordinate descent procedure, we **first choose a direction** (say the j-th co-ordinate), and then **find how much to move along that co-ordinate**.
- The step length z is obtained by considering the following one-dimensional problem :

$$\min_z \left[\lambda|\mathbf{w}_j + z| - \lambda|\mathbf{w}_j| + \sum_{i \in I(\mathbf{w}+z\mathbf{e}_j)} (b_i(\mathbf{w} + z\mathbf{e}_j))^2 - \sum_{i \in I(\mathbf{w})} (b_i(\mathbf{w}))^2 \right]$$

$$\approx \min_z \left[\lambda|\mathbf{w}_j + z| + \mathcal{L}'_j(0; \mathbf{w})z + \frac{1}{2}\mathcal{L}''_j(0; \mathbf{w})z^2 \right] \text{ (Taylor expansion w.r.t. } z) \quad (1)$$

Where

- $\mathcal{L}_j(z; \mathbf{w}) := \sum_{i \in I(\mathbf{w}+z\mathbf{e}_j)} (b_i(\mathbf{w} + z\mathbf{e}_j))^2$
- $\mathcal{L}'_j(0; \mathbf{w}) = -2 \sum_{i \in I(\mathbf{w})} y_i \mathbf{x}_{ij} (b_i(\mathbf{w}))$
- $\mathcal{L}''_j(0; \mathbf{w}) = \max(2 \sum_{i \in I(\mathbf{w})} x_{ij}^2, 10^{-12})$.

[1] More details in the Liblinear paper Appendix F
https://www.csie.ntu.edu.tw/~cjlin/papers/liblinear.pdf

Co-ordinate Descent Method

- The optimal step length, d, is obtained by minimizing equation (**??**) (which is the quadratic approximation using Taylor series) :

$$d = \begin{cases} -\frac{\mathcal{L}'_j(0;\mathbf{w})+\lambda}{\mathcal{L}''_j(0;\mathbf{w})} & \text{if } \mathcal{L}'_j(0;\mathbf{w}) + \lambda \leq \mathcal{L}''_j(0;\mathbf{w})\mathbf{w}_j \\ -\frac{\mathcal{L}'_j(0;\mathbf{w})-\lambda}{\mathcal{L}''_j(0;\mathbf{w})} & \text{if } \mathcal{L}'_j(0;\mathbf{w}) - \lambda \geq \mathcal{L}''_j(0;\mathbf{w})\mathbf{w}_j \\ -\mathbf{w}_j & \text{otherwise} \end{cases}$$

Co-ordinate Descent Method

- The optimal step length, d, is obtained by minimizing equation (**??**) (which is the quadratic approximation using Taylor series) :

$$d = \begin{cases} -\frac{\mathcal{L}'_j(0;\mathbf{w})+\lambda}{\mathcal{L}''_j(0;\mathbf{w})} & \text{if } \mathcal{L}'_j(0;\mathbf{w}) + \lambda \leq \mathcal{L}''_j(0;\mathbf{w})\mathbf{w}_j \\ -\frac{\mathcal{L}'_j(0;\mathbf{w})-\lambda}{\mathcal{L}''_j(0;\mathbf{w})} & \text{if } \mathcal{L}'_j(0;\mathbf{w}) - \lambda \geq \mathcal{L}''_j(0;\mathbf{w})\mathbf{w}_j \\ -\mathbf{w}_j & \text{otherwise} \end{cases}$$

- How about the optimality condition for the j-th co-ordinate

Co-ordinate Descent Method

- The optimal step length, d, is obtained by minimizing equation (**??**) (which is the quadratic approximation using Taylor series) :

$$d = \begin{cases} -\frac{\mathcal{L}'_j(0;\mathbf{w})+\lambda}{\mathcal{L}''_j(0;\mathbf{w})} & \text{if } \mathcal{L}'_j(0;\mathbf{w}) + \lambda \leq \mathcal{L}''_j(0;\mathbf{w})\mathbf{w}_j \\ -\frac{\mathcal{L}'_j(0;\mathbf{w})-\lambda}{\mathcal{L}''_j(0;\mathbf{w})} & \text{if } \mathcal{L}'_j(0;\mathbf{w}) - \lambda \geq \mathcal{L}''_j(0;\mathbf{w})\mathbf{w}_j \\ -\mathbf{w}_j & \text{otherwise} \end{cases}$$

- How about the optimality condition for the j-th co-ordinate
- The optimality condition along $\mathbf{w}_j$ is given by the following :

$$\begin{array}{ll} \mathcal{L}'_j(0;\mathbf{w}) + \lambda = 0 & \text{if } \mathbf{w}_j > 0 \\ \mathcal{L}'_j(0;\mathbf{w}) - \lambda = 0 & \text{if } \mathbf{w}_j < 0 \\ -\lambda \leq \mathcal{L}'_j(0;\mathbf{w}) \leq \lambda & \text{if } \mathbf{w}_j = 0 \end{array} \tag{2}$$

Algorithm for Co-ordinate Descent

Require: Training data $(\mathbf{X}, \mathbf{y})$ and initialize $\mathbf{w} = \mathbf{0}$
Ensure: Learnt weight vector $\mathbf{w}^*$

1: **while** $\mathbf{w}$ is not optimal **do**
2: **for** j=1,. . . ,D **do** ▷ Iterate over co-ordinates
3: Solve the following quadratic approximation and find step length d

$$\min_z \left[\lambda|\mathbf{w}_j + z| - \lambda|\mathbf{w}_j| + \sum_{i \in I(\mathbf{w}+z\mathbf{e}_j)} (b_i(\mathbf{w} + z\mathbf{e}_j))^2 - \sum_{i \in I(\mathbf{w})} (b_i(\mathbf{w}))^2\right]$$

4: $\mathbf{w}_j = \mathbf{w}_j + d$ [2]
5: Check if the optimality condition is satisfied for the j-th co-ordinate
6: If satisfied, remove that variable from optimization, else continue
7: **end for**
8: **end while**

[2] Before this step, one needs to check the sufficient decrease condition, skipped here for simplicity

Explicit Non-linear Feature Maps and Bochner's Theorem

NIPS 2017 Test of Time Award

Random Features for Large-Scale Kernel Machines

Ali Rahimi and Ben Recht

Abstract

To accelerate the training of kernel machines, we propose to map the input data to a randomized low-dimensional feature space and then apply existing fast linear methods. Our randomized features are designed so that the inner products of the transformed data are approximately equal to those in the feature space of a user specified shift-invariant kernel. We explore two sets of random features, provide convergence bounds on their ability to approximate various radial basis kernels, and show that in large-scale classification and regression tasks linear machine learning algorithms that use these features outperform state-of-the-art large-scale kernel machines.

- Video from NIPS 2017 with Award talk at - `https://www.youtube.com/watch?v=Qi1Yry33TQE`
- A workshop at Princeton in Febuary 2019 (with videos) - `http://www.math.ias.edu/tml/dlasagenda` arising out the points made in above talk

Fourier transforms

- **Fourier transform** $S(\omega)$ of a function $f(x)$,

$$S(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx$$

where

- i is the imaginary number with $i^2 = -1$ and $i^0 = 1$
- ω is a frequency

Fourier transforms

- **Fourier transform** $S(\omega)$ of a function $f(x)$,

$$S(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx$$

  where
  - i is the imaginary number with $i^2 = -1$ and $i^0 = 1$
  - ω is a frequency
- **Inverse Fourier transform** $f(x)$ of spectral density $S(\omega)$,

$$f(x) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i x \omega} d\omega$$

Fourier transforms

- **Fourier transform** $S(\omega)$ of a function $f(x)$,

$$S(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx$$

  where
  - i is the imaginary number with $i^2 = -1$ and $i^0 = 1$
  - ω is a frequency
- **Inverse Fourier transform** $f(x)$ of spectral density $S(\omega)$,

$$f(x) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i x \omega} d\omega$$

- Euler's identity helps compute Fouriers in practice

$$e^{ix} = \underbrace{\cos x}_{\text{real part}} + \underbrace{i \cdot sinx}_{\text{complex part}}$$

Fourier transforms

- **Fourier transform** $S(\omega)$ of a function $f(x)$,

$$S(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx$$

  where
  - i is the imaginary number with $i^2 = -1$ and $i^0 = 1$
  - ω is a frequency
- **Inverse Fourier transform** $f(x)$ of spectral density $S(\omega)$,

$$f(x) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i x \omega} d\omega$$

- Euler's identity helps compute Fouriers in practice

$$e^{ix} = \underbrace{\cos x}_{\text{real part}} + \underbrace{i \cdot sinx}_{\text{complex part}}$$

- Hence,

$$e^{-2\pi i x \omega} = \cos(2\pi x \omega) - i \sin(2\pi x \omega)$$
$$e^{2\pi i x \omega} = \cos(2\pi x \omega) + i \sin(2\pi x \omega)$$

Bochner's Theorem

Let the input space $\mathcal{X} = \mathbb{R}^D$

- A kernel $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is said to be translation invariant if it satisfies $k(x, x') := \psi(x - x')$ for a positive definite function $\psi(.)$ (with one argument) on $\mathbb{R}^D$.
- Bochner theorem (stated in somewhat simpler terms below) gives a complete characterization of a kernel function

Bochner's Theorem

Let the input space $\mathcal{X} = \mathbb{R}^D$

- A kernel $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is said to be translation invariant if it satisfies $k(x, x') := \psi(x - x')$ for a positive definite function $\psi(.)$ (with one argument) on $\mathbb{R}^D$.
- Bochner theorem (stated in somewhat simpler terms below) gives a complete characterization of a kernel function

Theorem (Bochner)

A continuous function $\psi : \mathbb{R}^D \mapsto \mathbb{R}$ is positive definite if and only if it is the inverse fourier transform of a probability density function.

$$\psi(\tau) = \int_{-\infty}^{\infty} p(\omega) e^{2\pi i \omega^T \tau} d\omega \qquad (p(\omega) \text{ is a probability density function})$$

Bochner's Theorem

Let the input space $\mathcal{X} = \mathbb{R}^D$

- A kernel $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is said to be translation invariant if it satisfies $k(x, x') := \psi(x - x')$ for a positive definite function $\psi(.)$ (with one argument) on $\mathbb{R}^D$.
- Bochner theorem (stated in somewhat simpler terms below) gives a complete characterization of a kernel function

Theorem (Bochner)

A continuous function $\psi : \mathbb{R}^D \mapsto \mathbb{R}$ is positive definite if and only if it is the inverse fourier transform of a probability density function.

$$\psi(\tau) = \int_{-\infty}^{\infty} p(\omega) e^{2\pi i \omega^T \tau} d\omega \qquad (p(\omega) \text{ is a probability density function})$$

What other characterizations of kernels we have seen already?

Bochner's Theorem

Let the input space $\mathcal{X} = \mathbb{R}^D$

- A kernel $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is said to be translation invariant if it satisfies $k(x, x') := \psi(x - x')$ for a positive definite function $\psi(.)$ (with one argument) on $\mathbb{R}^D$.
- Bochner theorem (stated in somewhat simpler terms below) gives a complete characterization of a kernel function

Theorem (Bochner)

A continuous function $\psi : \mathbb{R}^D \mapsto \mathbb{R}$ is positive definite if and only if it is the inverse fourier transform of a probability density function.

$$\psi(\tau) = \int_{-\infty}^{\infty} p(\omega) e^{2\pi i \omega^T \tau} d\omega \qquad (p(\omega) \text{ is a probability density function})$$

What other characterizations of kernels we have seen already?

- Existence of a feature map $\phi(.)$ and a feature space $\mathcal{H}$ (recall lecture 1)
- Postive definiteness (recall lecture 2)

Simplified Representation for Symmetric Distributions

- Assume symmetric distribution $p(\omega) = p(-\omega)$
- Euler's identity $e^{\pm ix} = \cos x \pm i \sin x$
- Sine identity $\sin(-x) = -\sin(x)$
- Then we can solve the inverse Fourier representation on previous slide as follows

Simplified Representation for Symmetric Distributions

- Assume symmetric distribution $p(\omega) = p(-\omega)$
- Euler's identity $e^{\pm ix} = \cos x \pm i \sin x$
- Sine identity $\sin(-x) = -\sin(x)$
- Then we can solve the inverse Fourier representation on previous slide as follows

$$
\begin{aligned}
\psi(\tau) &= \int_{-\infty}^{\infty} p(\omega) e^{2\pi i \tau \omega} d\omega \\
&= \int_{-\infty}^{\infty} p(\omega) \cos(2\pi \omega^T \tau) d\omega + \int_{-\infty}^{\infty} i \cdot p(\omega) \sin(2\pi \omega^T \tau) d\omega \\
&= \mathbb{E}_\omega[\cos(2\pi \omega^T \tau)] + \int_{-\infty}^{0} i \cdot p(\omega) \sin(2\pi \omega^T \tau) d\omega + \int_{0}^{\infty} i \cdot p(\omega) \sin(2\pi \omega^T \tau) d\omega \\
&= \mathbb{E}_\omega[\cos(2\pi \omega^T \tau)] + \int_{0}^{\infty} i \cdot p(\omega) \sin(2\pi (-\omega)^T \tau) d\omega + \int_{0}^{\infty} i \cdot p(\omega) \sin(2\pi \omega^T \tau) d\omega \\
&= \mathbb{E}_\omega[\cos(2\pi \omega^T \tau)] - \int_{0}^{\infty} i \cdot p(\omega) \sin(2\pi \omega^T \tau) d\omega + \int_{0}^{\infty} i \cdot p(\omega) \sin(2\pi \omega^T \tau) d\omega \\
&= \mathbb{E}_\omega[\cos(2\pi \omega^T \tau)]
\end{aligned}
$$

Bochner's Theorem - Consequence I

Given an integrable function $\psi(\tau)$, i.e. $\int_{\mathbb{R}^D} |\psi(\tau)| < \infty$, compute its Fourier transform as follows

Checking for positive definiteness of a function

$$p(\omega) = \int_{-\infty}^{\infty} \psi(\tau) e^{-2\pi i \omega^T \tau} d\tau,$$

If $p(\omega)$ is non-negative $\forall \omega \in \mathbb{R}^D$, then $\psi(.)$ is a positive definite function, and hence $k(\mathbf{x}, \mathbf{x}')$ is a kernel

Bochner's Theorem - Consequence II

Recall from our notation $\psi(\tau) := \psi(\mathbf{x} - \mathbf{x}') := k(\mathbf{x}, \mathbf{x}')$

$$
\begin{aligned}
\psi(\tau) &= \mathbb{E}_\omega[\cos(2\pi\omega^T\tau)] \\
&= \mathbb{E}_\omega[\cos(2\pi\omega^T(\mathbf{x} - \mathbf{x}'))] \\
&\approx \frac{1}{L}\sum_{\ell=1}^{L}[\cos(2\pi\omega^T(\mathbf{x} - \mathbf{x}'))] \\
&= \frac{1}{L}\sum_{\ell=1}^{L}[\cos(2\pi\omega^T\mathbf{x})\cos(2\pi\omega^T\mathbf{x}') + \sin(2\pi\omega^T\mathbf{x})\sin(2\pi\omega^T\mathbf{x}')] \\
&= \frac{1}{L}\sum_{\ell=1}^{L}\left\langle[\cos(2\pi\omega^T\mathbf{x}), \sin(2\pi\omega^T\mathbf{x})], [\cos(2\pi\omega^T\mathbf{x}'), \sin(2\pi\omega^T\mathbf{x}')]\right\rangle \\
&= \langle\phi(x), \phi(x')\rangle
\end{aligned}
$$

where $\phi(\mathbf{x}) = \frac{1}{\sqrt{L}}[\ldots, \cos(2\pi\omega_\ell^T\mathbf{x}), \sin(2\pi\omega_\ell^T\mathbf{x}), \ldots] \in \mathbb{R}^{2L}$ is the approximate non-linear feature map.

Algorithm for Computing Random Fourier Features

Require: Training data $\{\mathbf{x}_i\}_{i=1}^N$ such that $\mathbf{x}_i \in \mathbb{R}^D$and a kernel $k(,.)$ in the form of translation invariant function $\psi(\tau))$

Ensure: Non-linear Feature maps $\phi(\mathbf{x})$ for instance $\mathbf{x} \in \mathbf{X}$

1: Compute Fourier transform of the kernel to get the probability density $p(w) = \int_{-\infty}^{\infty} \psi(\tau) e^{-2\pi i \omega^T \tau} d\tau$

2: Draw L i.i.d. samples $\omega_1, \ldots, \omega_L \in \mathbb{R}^D$ from the distribution p

3: For each training instance $\mathbf{x}_i$, return the corresponding feature map $\phi(\mathbf{x}_i) = \frac{1}{\sqrt{L}}[\ldots, \cos(2\pi\omega_\ell^T \mathbf{x}), \sin(2\pi\omega_\ell^T \mathbf{x}), \ldots]$

One can then run any linear method such that discussed in first part of the lecture by replacing $\mathbf{x}$ by $\phi(\mathbf{x})$

Summary

- Kernel methods provide a flexible way to implcitly represent data in high dimensional spaces
- However, for big data problems, computations involving kernel matrix can become computationally inefficient
- To tackle those, we looked at two scenarios
  - Large-scale linear classification when the input data is inherently high dimensional (using co-ordinate decent as an example)
  - Scaling up Kernel methods by explicit feature mappings via Bochner's Theorem