Machine Learning in Image Analysis
Day 1

Anirban Mukhopadhyay
Zuse Institute Berlin

Organization

- Why Machine Learning for Image Analysis
- Image Analysis Perspective
- Types of Model
- Empirical Risk Minimization
- Essentials of convexity (Sets, Function, Operations)
- Intro to linear SVM
- Cutting Plane Method to solve linear SVM

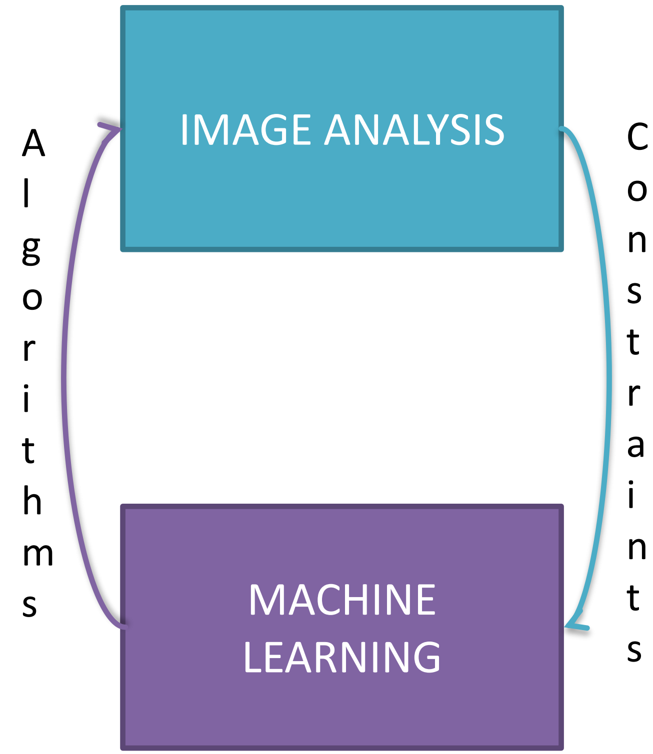
Machine Learning

- Field of study that gives computers the ability to learn without being explicitly programmed
  - Arthur Samuel, 1959 / Wiki definition

Supervised	Semi-Supervised	Unsupervised
Generative	Metric Learning	Clustering
Discriminative	…	…

Why ML for IA?

- IA: Infer information from visual data
  - Segmentation
  - Registration
  - Recognition
  - Image Guided Therapy …
- Large variations and complexity
  - No analytical solution
- Resort to ML



IA problems that can benefit from ML

- NP-Hard (ex: scene matching)
- Ill-defined (ex: 3D reconstruction from a single image)
- Right answer is subjective (ex: segmentation)
- Hard to model (ex: scene classification)

- ML uses statistical reasoning to find approximate solutions for tackling the above difficulties.

Formulating and Evaluating IA problems as ML

- Topic of Day 3
  - Read 4 sample papers (Medical Image Analysis + Computer Vision)
  - Critically analyze the contributions
  - It’s not about blind accuracy plot w.r.t. different off-the-shelf methods … there are many more nuances

- List of papers: www.zib.de/MLIA

Image Analysis Perspective

- Given visual data x, infer world state y
  - Discrete -> Classification
  - Continuous -> Regression

Image Analysis Perspective

- Given visual data x, infer world state y
  - Discrete -> Classification
  - Continuous -> Regression

- Components of the solution
  - Model
  - Learning Algorithm
  - Inference Algorithm

Components of the solution (Contd.)

- Model: Mathematically relate visual data x with world state y

Components of the solution (Contd.)

- Model: Mathematically relate visual data x with world state y
- Learning Algo: Fit parameters θ using paired training examples (x_i, y_i)

Components of the solution (Contd.)

- Model: Mathematically relate visual data x with world state y
- Learning Algo: Fit parameters θ using paired training examples (x_i, y_i)
- Inference Algo: Take a new observation x and use learnt model to predict world state y

Types of Model

	Generative	Discriminative
Local	Max. Likelihood	Empirical Risk Minimization
Local+Prior	MAP	Support Vector Machines
Model Averaging	Bayesian	Maximum Entropy Discrimination

Choosing one over the other

- No Definitive Answer.

Choosing one over the other

- No Definitive Answer.
- Some considerations:
  - Inference is generally simpler with discriminative

Choosing one over the other

- No Definitive Answer.
- Some considerations:
  – Inference is generally simpler with discriminative
  – Image data are generally much higher dimensional than world state – modeling is costly

Choosing one over the other

- No Definitive Answer.
- Some considerations:
  - Inference is generally simpler with discriminative
  - Image data are generally much higher dimensional than world state – modeling is costly
  - If wishing to build information about the data generation process – generative

Choosing one over the other

- No Definitive Answer.
- Some considerations:
  - Inference is generally simpler with discriminative
  - Image data are generally much higher dimensional than world state – modeling is costly
  - If wishing to build information about the data generation process – generative
  - If missing data in training/ testing – generative

Choosing one over the other

- No Definitive Answer.
- Some considerations:
  - Inference is generally simpler with discriminative
  - Image data are generally much higher dimensional than world state – modeling is costly
  - If wishing to build information about the data generation process – generative
  - If missing data in training/ testing – generative
  - Expert knowledge incorporation as prior - generative

Empirical Risk Minimization

Quantification: Performance is Quantified by a loss function
Most Importantly: Generalize to unseen data – this is where optimization in ML is different from any other field
Idea: Avoid over-fitting by penalizing complex models

Empirical Risk Minimization

Quantification: Performance is Quantified by a loss function
Most Importantly: Generalize to unseen data – this is where optimization in ML is different from any other field
Idea: Avoid over-fitting by penalizing complex models

Training Data: $\{x_1, x_2, ..., x_m\}$
Training Labels: $\{y_1, y_2, ..., y_m\}$
Learn a vector: w

$$\underset{w}{\text{minimize}} \quad \boxed{\lambda\omega(w)} + \boxed{\frac{1}{m}\sum_{i=1}^{m} l(x_i, y_i, w)}$$

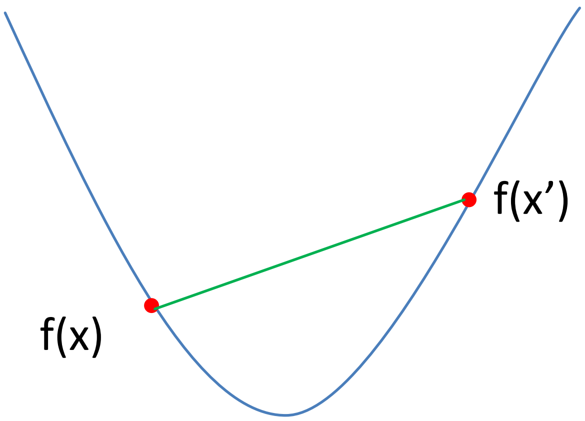
Regularizer Risk

ML directions

- Engineering part: Choose a loss and a regularizer based on your problem and go on .
- Optimization Part: If EMP can be turned into a convex problem…u can manage lots of things

- Our Focus: Intuition rather than rigor

Convex Function



- A function f is convex if and only if, for all x, x' and $\lambda \in (0, 1)$

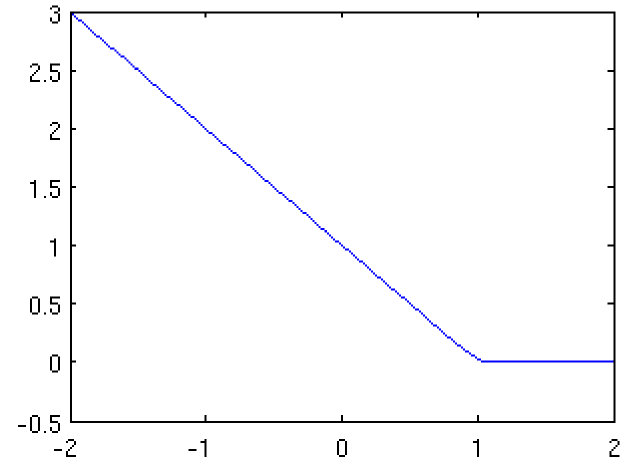
$$f(\lambda x + (1-\lambda)x') \leq \lambda f(x) + (1-\lambda) f(x')$$

Essential Convex Functions

Negative Entropy: $f(x) = xlogx + (1-x)log(1-x)$

Un-normalize Negative Entropy: $f(x,y) = xlogx + ylogy - x - y$

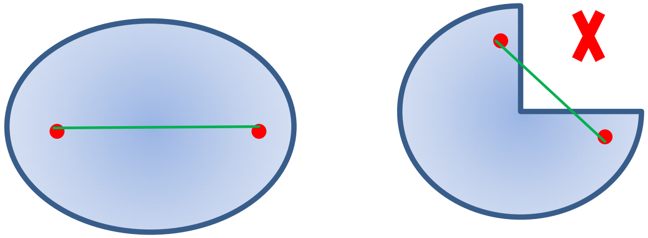
Hinge Loss: f(x) = max(0,1-x)



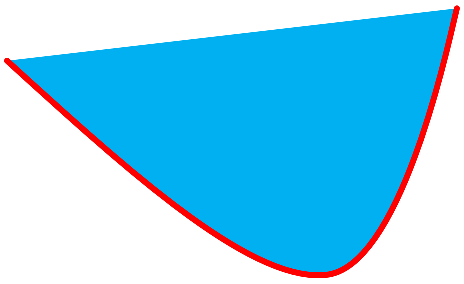
Convex set

- Set C is convex if and only if

$$\lambda x + (1-\lambda)x' \in C$$

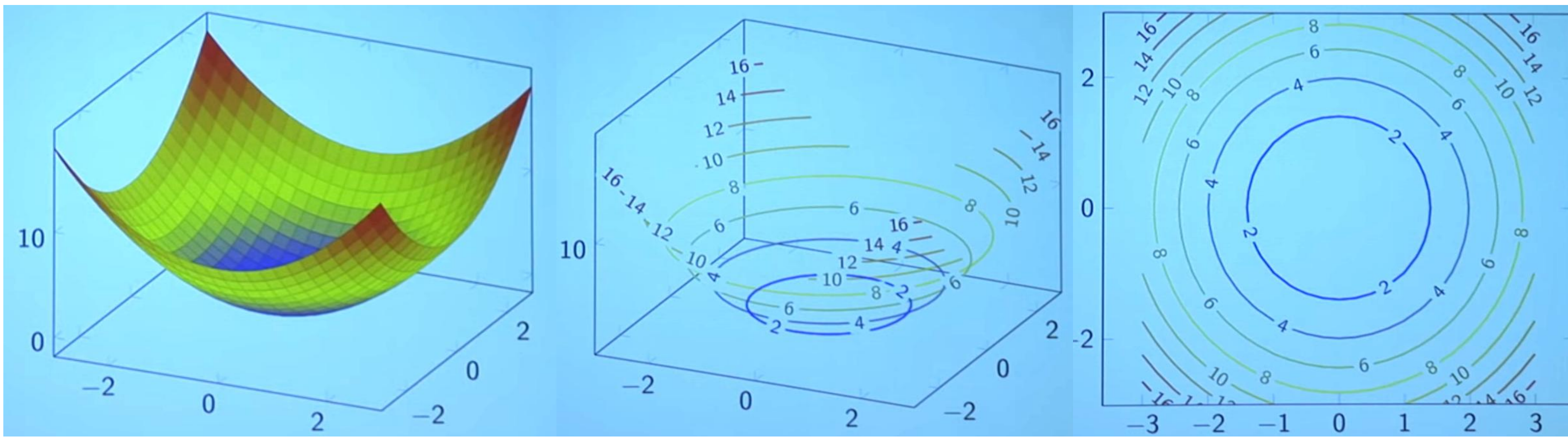


- If a function is convex, all its level sets are convex

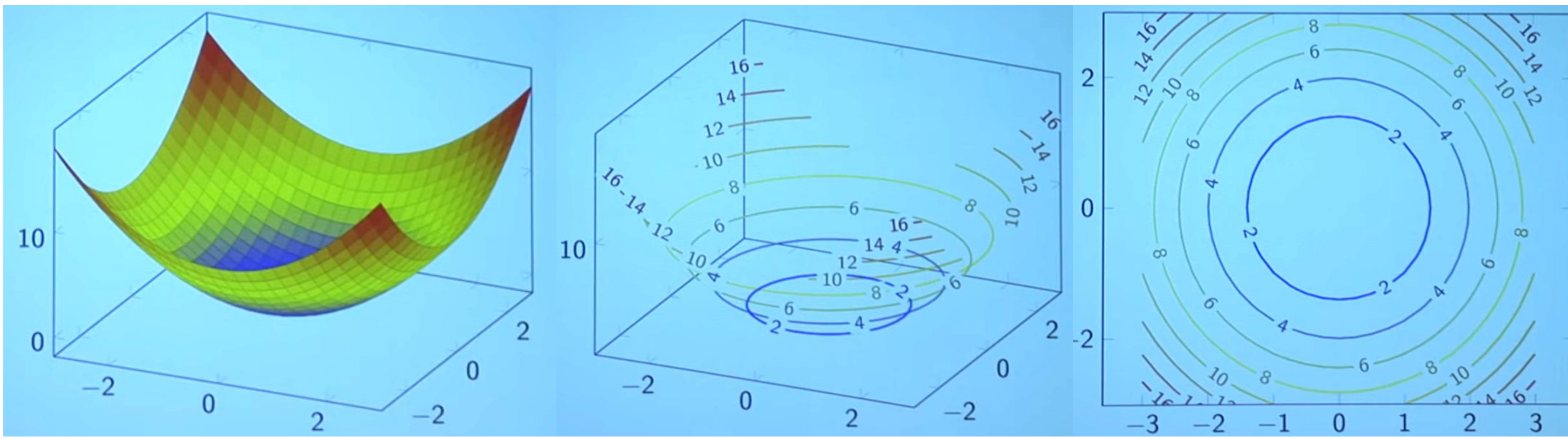


Function is convex if and only if epigraph is a convex set

Level Set Example



Level Set Example



BUT the converse is not true (quasi-convex)

Essential operations that preserve convexity

- Set Operations
  - Intersection of Convex Sets
  - Image of Convex Set under Linear Transf.
  - Inv. Image of Convex Set under Linear Transf.

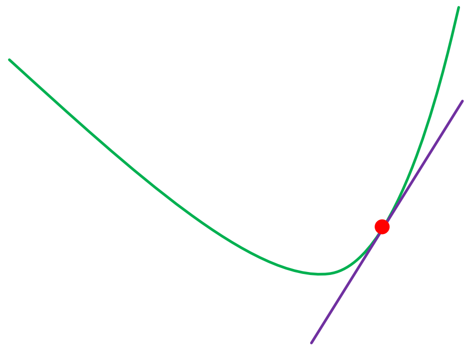
Essential operations that preserve convexity

- Set Operations
  - Intersection of Convex Sets
  - Image of Convex Set under Linear Transf.
  - Inv. Image of Convex Set under Linear Transf.
- Function Operations
  - Linear Combination with non-negative weights
  - Point wise Maximum
  - Projection along a direction
  - Composition with affine function

First Order Properties

- First order Taylor Approx. Globally lower bounds a function

$$f(x) \geq f(x') + < x - x', \nabla f(x') >$$

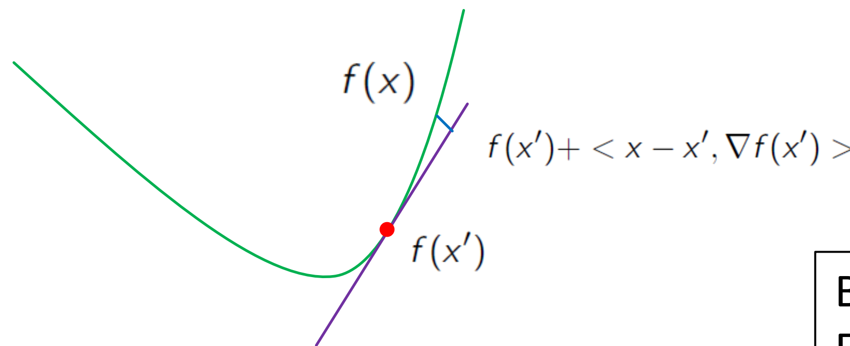


Where ever u go, the line will never intersect the function anywhere else apart from the red point

Bregman Divergence

$$\Delta_f(x, x') = f(x) - f(x') - < x - x', \nabla f(x') >$$

As given by the function, how far away is x from x'



Bcoz 1^{st} order Taylor Expansion is global lower bound, f(x) is larger than the other

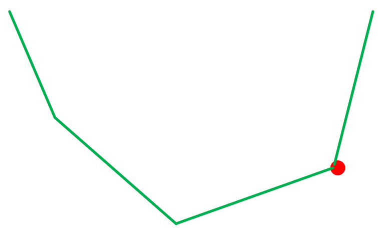
- 2 Popular flavors
  - Euclidean Distance Squared
  - Unnormalized Relative Entropy

Identifying the Minima

Given a smooth (differentiable) convex function f

$$\nabla f(x) = 0$$

- What if function is non-smooth?

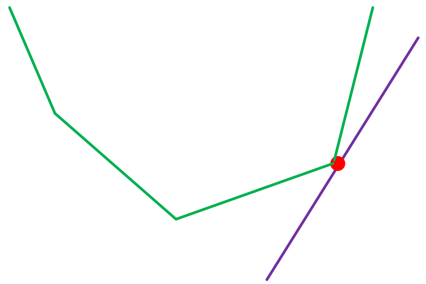


Identifying the Minima

Given a smooth (differentiable) convex function f

$$\nabla f(x) = 0$$

- What if function is non-smooth?

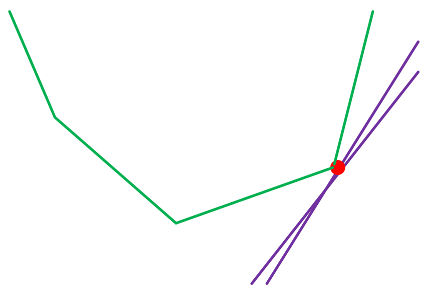


Identifying the Minima

Given a smooth (differentiable) convex function f

$$\nabla f(x) = 0$$

- What if function is non-smooth?



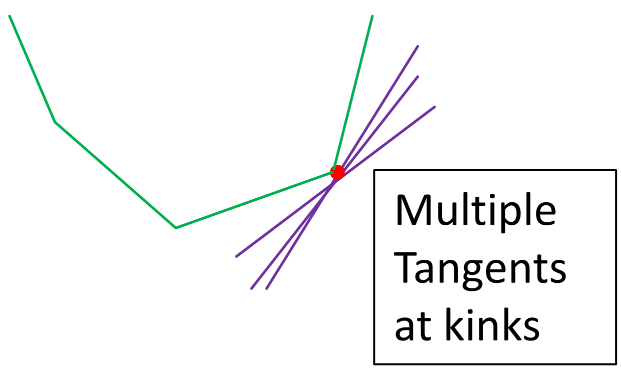
Identifying the Minima

Given a smooth (differentiable) convex function f

$$\nabla f(x) = 0$$

- What if function is non-smooth?

Subgradients - to the rescue



Identifying the Minima

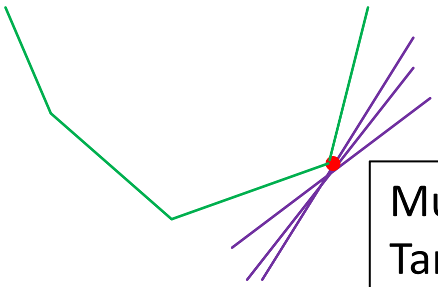
Given a smooth (differentiable) convex function f

$$\nabla f(x) = 0$$

- What if function is non-smooth?

Subgradients - to the rescue

Even in non-differentiable places, subgradient will always exist
You can always draw at least one tangent line



Multiple Tangents at kinks

Identifying the Minima

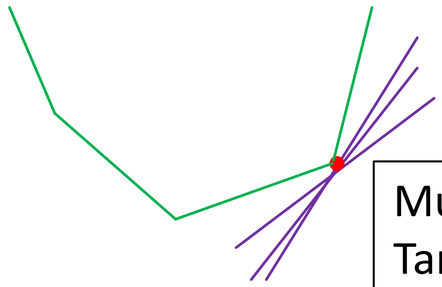
Given a smooth (differentiable) convex function f

$$\nabla f(x) = 0$$

- What if function is non-smooth?

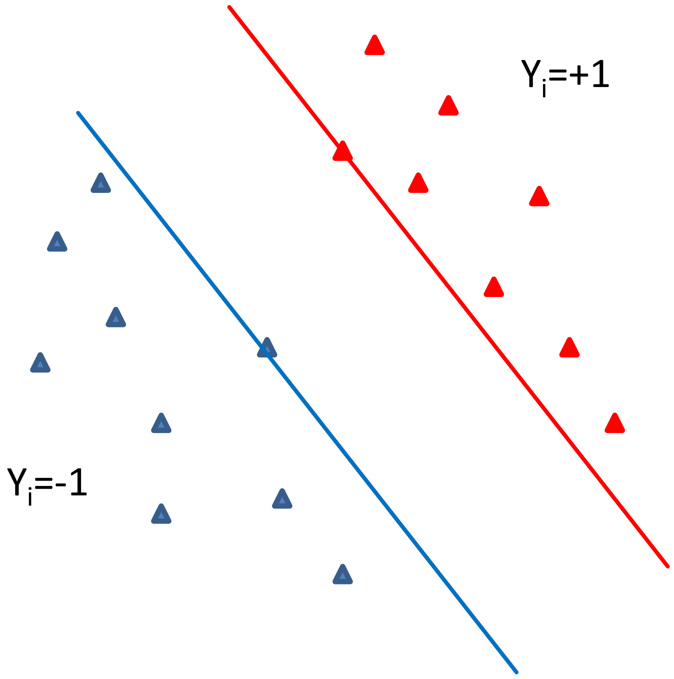
Remarkable property: A convex function is at least sub-differentiable everywhere

Even in non-differentiable places, subgradient will always exist
You can always draw at least one tangent line

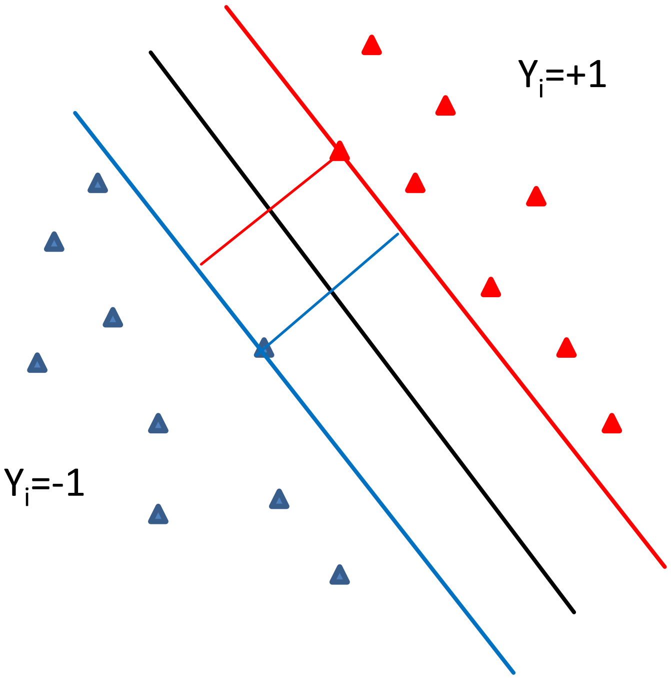


Multiple Tangents at kinks

Solving linear SVM



Solving linear SVM



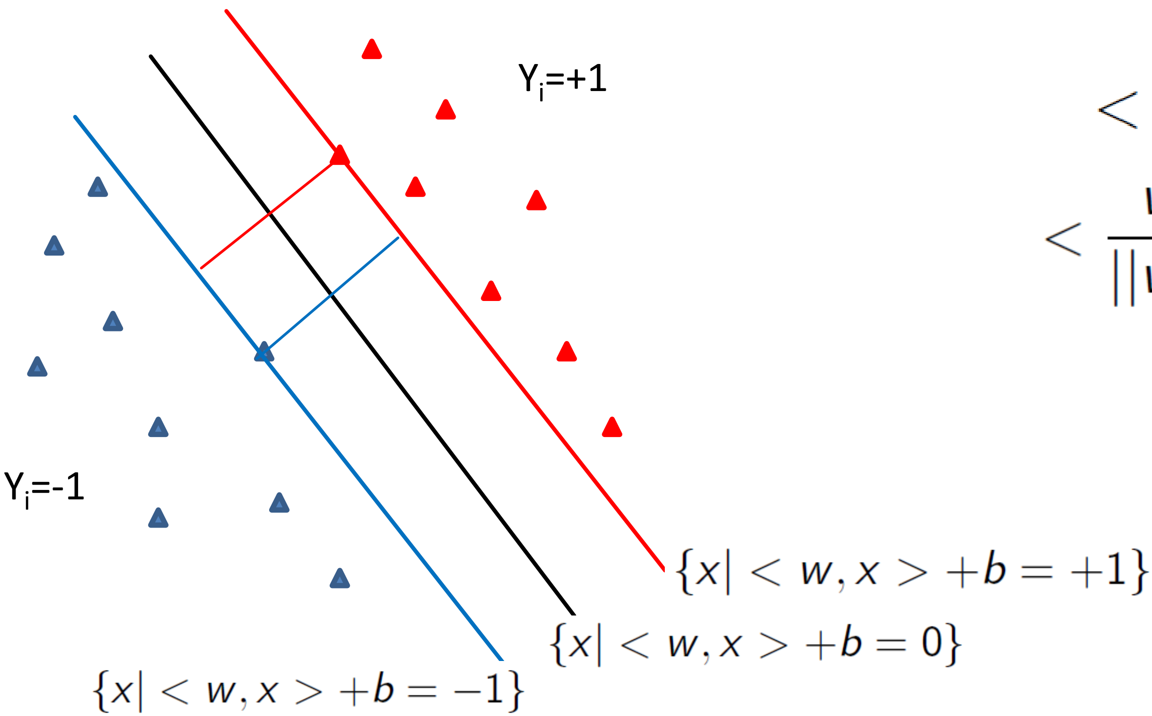
- Maximally non-committal hyperplane

Solving linear SVM

- Maximally non-committal hyperplane

$$< w, x_1 - x_2 >= 2$$

$$< \frac{w}{||w||}, x_1 - x_2 >= \frac{2}{||w||}$$



Optimization Problem

$$\underset{w,b}{\text{maximize}} \frac{2}{||w||} \quad \text{s. t.} \quad y_i(< w, x_i > +b) \geq 1, \forall i$$

Or

$$\underset{w,b}{\text{minimize}} \frac{1}{2}||w||^2 \quad \text{s. t.} \quad y_i(< w, x_i > +b) \geq 1, \forall i$$

More general ML problem

- Data is not exactly linearly separable
- Introduce slack variable

$$\underset{w,b,\xi}{\text{minimize}} \frac{1}{2}||w||^2 \quad \text{s. t.} \quad y_i(< w, x_i > +b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i$$

Slack Issues

- No control over slack variable, being $\xi_i \geq 0$
- Can go to infinity and find some useless solution

Slack Issues

- No control over slack variable, being $\xi_i \geq 0$
- Can go to infinity and find some useless solution
- Standard Solution: Penalize slack variables
  - Ensures nice classification for most of the points
  - Ready to pay the price for hopeless ones

$$\underset{w,b,\xi}{\text{minimize}} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i \quad \text{s. t.} \quad y_i(< w, x_i > + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i$$

Slack Issue Contd.

$$\underset{w,b,\xi}{\text{minimize}} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i \quad \text{s. t.} \quad y_i(< w, x_i > + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i$$

Slack Issue Contd.

$$\underset{w,b,\xi}{\text{minimize}} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i \quad \text{s. t.} \quad y_i(< w, x_i > +b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i$$

Or

$$\underset{w,b,\xi}{\text{minimize}} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i \quad \text{s. t.} \quad \xi_i \geq 1 - y_i(< w, x_i > +b), \xi_i \geq 0, \forall i$$

By standard optim. trick

$$\underset{w,b,\xi}{\text{minimize}} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i \quad \text{s. t.} \quad \xi_i \geq 1 - y_i(< w, x_i > + b), \xi_i \geq 0, \forall i$$

$$\Downarrow$$

$$\underset{w,b}{\text{minimize}} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m} max(0, 1 - y_i(< w, x_i > + b))$$

By standard optim. trick

$$\underset{w,b,\xi}{\text{minimize}} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i \quad \text{s. t.} \quad \xi_i \geq 1 - y_i(< w, x_i > +b), \xi_i \geq 0, \forall i$$

$$\Downarrow$$

$$\underset{w,b}{\text{minimize}} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m} max(0, 1 - y_i(< w, x_i > +b))$$

- Minimize squared Norm (want to have small w vectors)
- Hinge Loss (Risk Minimizer)

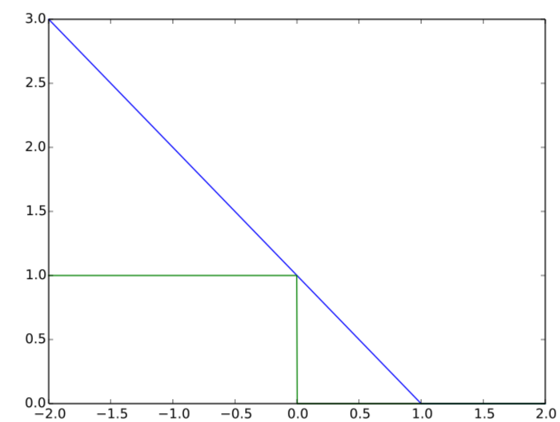
Loss Choices

$$\underset{w,b}{\text{minimize}} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m} max(0, 1 - y_i(< w, x_i > + b))$$

Regularizer

Risk

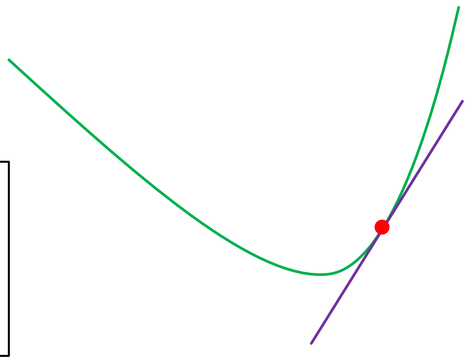
- Binary Loss
  - If correct, Nothing
  - If misclassification, unit loss
- But it is a nasty non-convex one, so take a convex upper bound e.g. Hinge Loss



Remember: First Order Properties

- First order Taylor Approx. Globally lower bounds a function

$$f(x) \geq f(x') + < x - x', \nabla f(x') >$$



Lower bound is piecewise linear – can use any LP solver to get some optimum

Where ever u go, the line will never intersect the function anywhere else apart from the red point

Cutting Plane method

- Idea: Localize your function
- Given:
  - black box which can calculate function value and gradient at any given point
  - Lower bound of the function (usually 0 for Regul. Risk Minimization)
- Remember: First order Taylor expansion globally lower bounds the function

Cutting Plane Method Visual

- Function resides in shaded area
- Refinement: Every time, we take a chunk out of the shaded by taking Taylor expansion

Check out the Board

More on Cutting Plane (CP)

- CP methods work by forming piecewise linear lower bound

$$J(w) \geq J_t^{CP}(w) = max_{1 \leq i \leq t}\{J(w_{i-1}) + < w - w_{i-1}, \nabla J(w_{i-1}) >\}$$

- At each iteration t, set $w_{0 \ldots t\text{-}1}$ is augmented by

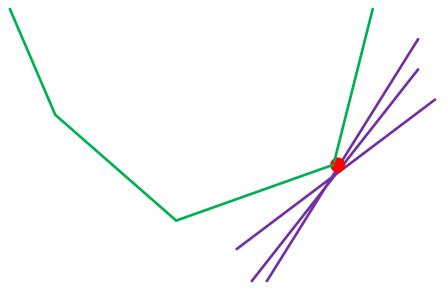
$$w_t = \underset{w}{\operatorname{argmin}} J_t^{CP}(w)$$

- Stop when gap

$$\epsilon_t = \min_{0 \leq i \leq t} J(w_i) - J_t^{CP}(w_t)$$

What if non-smooth function

- Cutting plane really does great in these situations, because it works on subgradients
- Choose any arbitrary subgradient and it will work.



Bundle Methods

- Stabilized Cutting Plane method (Always in practice)
- Add a regularizer to handle overfitting
  - Proximal: $w_t = \underset{w}{\text{argmin}}\{\frac{\xi_t}{2}||w - \hat{w}_{t-1}||^2 + J_t^{CP}(w)\}$
  - Trust region: $w_t = \underset{w}{\text{argmin}}\{J_t^{CP}(w) \quad \text{s. t.} \quad \frac{1}{2}||w - \hat{w}_{t-1}||^2 \leq K_t\}$
  - Level Set: $w_t = \underset{w}{\text{argmin}}\{\frac{1}{2}||w - \hat{w}_{t-1}||^2 \quad \text{s. t.} \quad J_t^{CP}(w) \leq \tau_t\}$

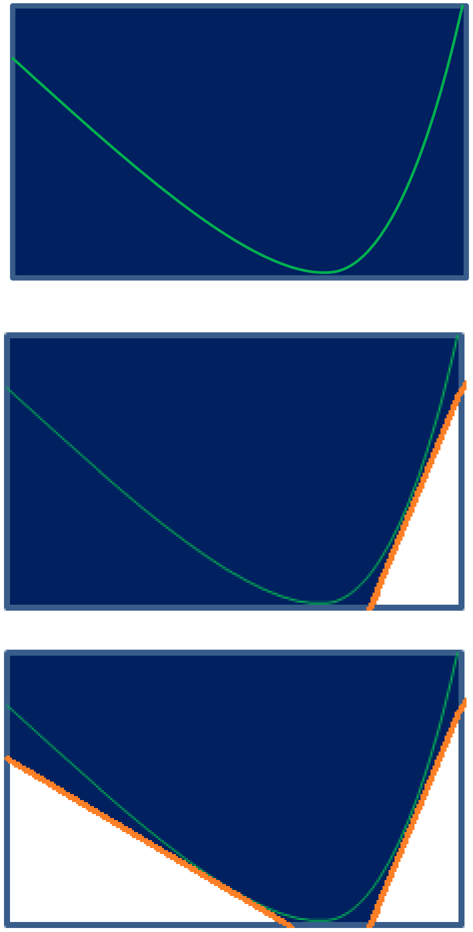
Quadratic in the gap calculation ensures convexity and unique minima

Referenes

- [PURDUE MLSS] SVN Vishwanathan Presentation
- Computer vision: models, learning and inference, Simon J.D. Prince, Cambridge University Press, 2012
- Optimization for Machine Learning, Sra, Nowozin, Wright, MIT Press, 2012
- Numerical Optimization, Nocedal, Wright, Springer, 1999
- Machine Learning in Computer Vision A Tutorial, Joshi, Cherian and Shivalingam, UMN

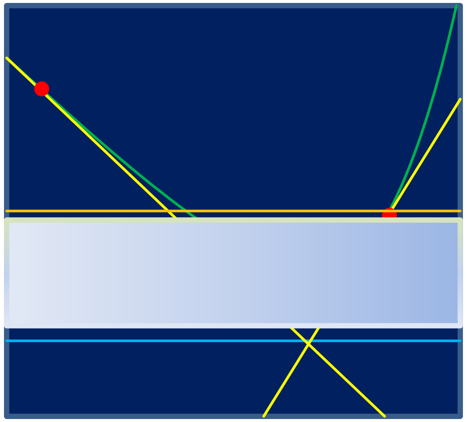
Cutting Plane Method Visual

- Function resides in checkerboard area
- Every time, we take a chunk out of the checkerboard by taking Taylor expansion



Turn Cutting Plane into Optimization

- Given: Green function and a second function that lies below green function
- Idea:
  - Minima of second function will always lie below blue function
  - Red points are always above true minima
  - Gap tells how far away u r from the optimum
- Solution: Optimize the gap to solve the problem



Understanding Bounds

Upper Bound

$$\exists c, \forall \epsilon > 0, \forall J \in F, \tau(\epsilon; J) \leq \frac{c}{\epsilon}$$

- $\exists c$: There is a constant
- $\forall \epsilon > 0$: Whatever constant u choose
- $\forall J \in F$: Whatever func. of this class
- $\tau(\epsilon; J) \leq \frac{c}{\epsilon}$: No. of steps the optim. needs for ε precision soln.

Lower Bound

$$\forall \epsilon > 0, \exists c, \exists J_\epsilon \in F, \quad \text{s. t.} \quad \tau(\epsilon, J_\epsilon) \geq \frac{c}{\epsilon}$$

- $\forall \epsilon > 0$: U fix an ε
- $\exists c, \exists J_\epsilon \in F$: I give u a const. and a bad func. belongs to F class
- $\tau(\epsilon, J_\epsilon) \geq \frac{c}{\epsilon}$: No. of steps the optim. needs for ε precision soln.

Turn Cutting Plane into Optimization

- Given: Green function and a second function that lies below green function
- Idea:
  - Minima of second function will always lie below blue function
  - Red points are always above true minima
  - Gap tells how far away u r from the optimum
- Solution: Optimize the gap to solve the problem