Intro to ML
Binary Classification with SVM

Evgeny Burnaev

Skoltech, Moscow, Russia

Outline

Machine Learning

- a broad subfield of Artificial Intelligence
- the mathematical discipline aimed at enabling machines (by computational methods) to improve at tasks with experience. The category includes deep learning
- data-driven $\rightarrow$ extracting patterns from data and based on mathematical statistics, numerical methods, optimization, probability theory, discrete analysis, geometry, etc.
- results of CS is used to analyze learning algorithms, their complexity, theoretical guarantees
- **Example**: predict a label of an image

Examples of Learning Tasks

- Image: annotation, segmentation, face recognition, OCR, face verification
- Autonomous technical systems (robots, cars)
- Medical diagnosis, fraud detection, network intrusions
- Playing games (chess, poker)
- Speech: recognition, synthesis, verification
- Text: topic modeling, spam detection

Math. ML Tasks

- **Dimensionality Reduction**: lower-dimensional features, preserving some properties of data
- **Regression**: predict some real-valued output variable for some input parameters (ship fuel consumption depending on weather conditions, route, etc.)
- **Classification**: set a label for each object (e.g. image classification)
- **Clustering**: partition objects into some "homogeneous" groups (e.g. divide documents into groups with similar topics)
- **Ranking**: rank objects according to some metric

What do we want?

- Algorithmic problems:
  - — more efficient and more accurate algorithms
  - — handle large-scale (dimensions, data volume) problems
  - — handle diverse types of data sources, including non-structured data, data on graphs, etc.

- Theoretical problems:
  - — what can be learned? under what conditions? restrictions?
  - — learning guarantees?
  - — learning algorithms performance?

Our aims

- Models and Algorithms
  — main algorithms and their efficiency
  — modern topics
- Theory
  — learning guarantees
  — analysis of algorithms
- Some applications (illustration)

Definitions

- **Example**: item, instance of data, related to some object
- **Features**: input value (input parameters, input vector, attributes, point) characterizing an object
- **Labels**: output value, categoric (classification) or real value (regression), associated to an object
- **Data**:
  - — training data (usually contains labels)
  - — test data (labels exist but not known)
  - — validation data (labeled, used for tuning of hyperparameters)

Learning Scenarious

- Settings

BATCH: a learner get full sample, learn a model and performs predictions for unseen points

ON-LINE: a learner receives one sample at a time and makes prediction for that sample

- Queries

ACTIVE: a learner can request the label of a point

PASSIVE: a learner always receives labeled points

Batch Settings

- **Unsupervised learning**: no labeled data
- **Supervised learning**: learn using labeled data for prediction on unseen points
- **Semi-supervised learning**: learn using labeled and unlabeled data for prediction on unseen points
- **Transduction**: uses labeled and unlabeled data for prediction on seen points

Example — Information Retrieval

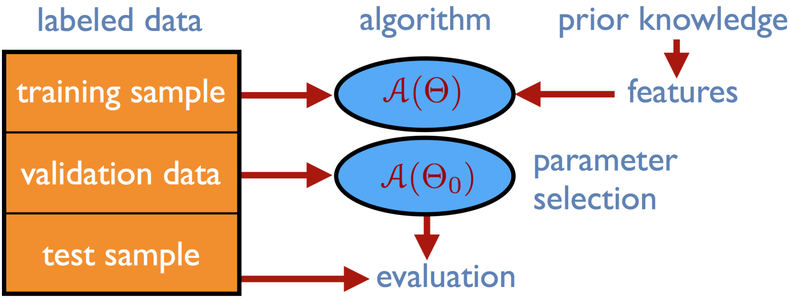
Information retrieval with relevance feedback

- User enters a query
- Machine returns sample document
- User labels the documents (relevant/not relevant)
- Machine selects most relevant documents from available

Relevance

- Obtaining labels require work from the user
- Obtaining documents is automatic (from database)
- Instances to be classified: documents of the database
- No need to know the classification function

Learning Scheme



Main Notions

- **Spaces**: input space X, output space Y
- **Loss function** $L : Y \times Y \to \mathbb{R}$
  - $L(y, \hat{y})$ is the error of predicting $\hat{y}$ instead of y
  - $0 - 1$ loss $L(y, \hat{y}) = 1_{y \neq \hat{y}}$ in case of binary classification
  - $L(y, \hat{y}) = (y - \hat{y})^2$ in case of regression with $Y \subseteq \mathbb{R}$
- **Hypothesis set** $H \subset Y^X$ is a subset of functions out of which the learner selects his hypothesis

  — represents a prior knowledge about the task at hand

  — depends on available features

Supervised Learning

- **Training data**: sample S of size m drawn $i.i.d.$ according to distribution D on $X \times Y$

$$S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$$

- **Problem** find hypothesis $h \in H$ with small generalization error
  - deterministic case: $y = f(x)$ is a deterministic function, only $x \sim D$
  - stochastic case: output is a probabilistic function of input, e.g. $y = f(x) + \varepsilon$

Errors

- **Generalization error**: for $h \in H$

$$R(h) = \mathbb{E}_{(x,y)\sim D}[L(h(x), y)]$$

- **Empirical error** for $h \in H$ and sample S

$$\hat{R}(h) = \frac{1}{m}\sum_{i=1}^{m} L(h(x_i), y_i)$$

- **Bayes error**

$$R^* = \inf_h R(h)$$

Errors

- **Noise**:
  - — in regression for any $x \in X$, L_2-loss and white noise model
$$h^* = \mathbb{E}(y|x)$$
  - — observe that
$$R^* = \mathrm{Var}(\varepsilon)$$

Learning ≠ Fitting

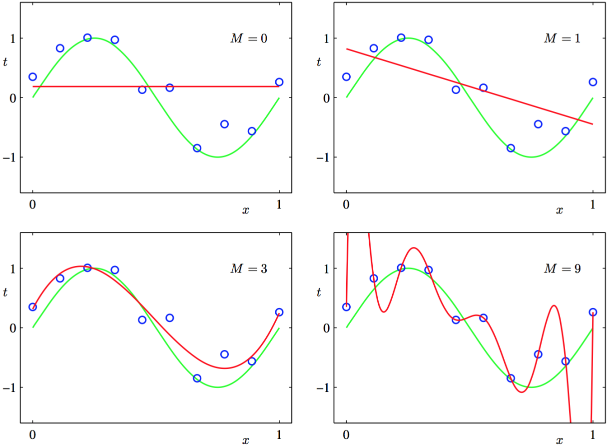


Figure : Notion of simplicity/complexity. How do we define complexity?

Empirical observations

- the best hypothesis on the sample may not be the best overall
- generalization is not memorization
- complex rules can provide poor predictions
- trade-off: complexity vs. sample size (underfitting/overfitting)

Learning ≠ Fitting

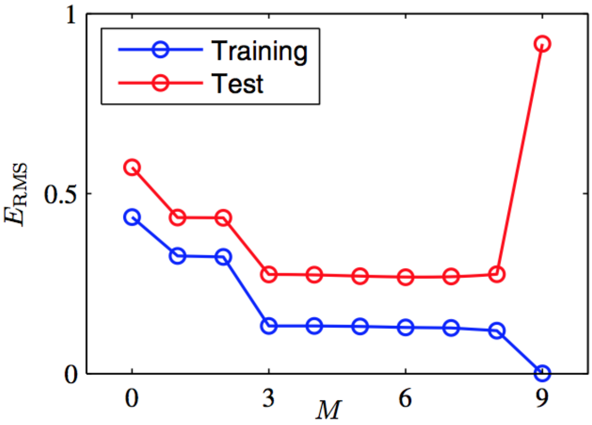


Figure : Notion of simplicity/complexity. How do we define complexity?

Empirical Risk Minimization

- Select Hypothesis set H
- Find hypothesis $h \in H$ minimizing empirical error

$$h = \arg\min_{h \in H} \hat{R}(h)$$

- H may be too complex
- Sample size may not be large enough

Structural Risk Minimization Principle

- Consider an infinite sequence of hypothesis sets ordered for inclusion

$$H_1 \subset H_2 \subset \ldots \subset H_n \subset \ldots$$

$$h = \arg \min_{h \in H_n,\, n \in \mathbb{N}} \hat{R}(h) + \mathrm{penalty}(H_n, m)$$

- Strong theoretical guarantees
- Typically computationally intensive

Families of Algorithms

- Empirical risk minimization (ERM)

$$h = \arg\min_{h \in H} \hat{R}(h)$$

- Structural Risk Minimization (SRM) for $H_n \subseteq H_{n+1}$

$$h = \arg\min_{h \in H_n,\, n \in \mathbb{N}} \hat{R}(h) + \mathrm{penalty}(H_n, m)$$

- Regularization-based algorithms

$$h = \arg\min_{h \in H} \hat{R}(h) + \lambda \|h\|^2,\ \lambda > 0$$

Problem Statement

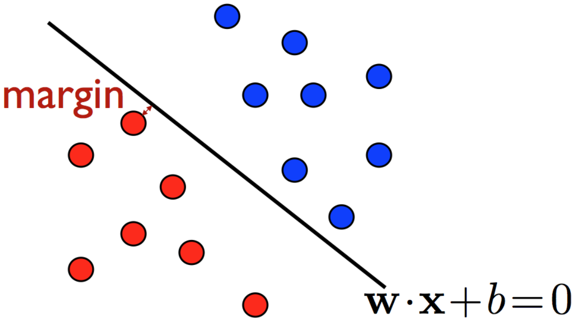
- **Training data**: sample drawn i.i.d. w.r.t. D on $X \subseteq \mathbb{R}^N$

$$S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \in \{X \times \{-1, +1\}\}^m$$

- **Problem**: find hypothesis $h : X \to \{-1, +1\}$ in H (classifier) with small generalization error $R(h)$

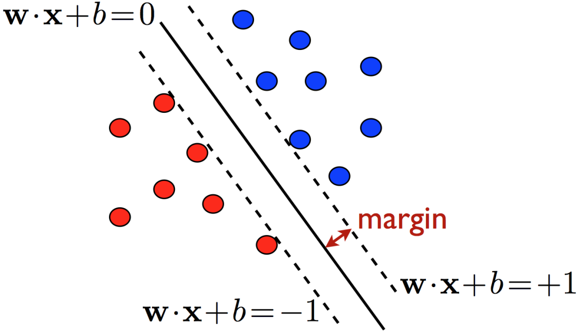
- First we consider linear classification (hyperplanes) if dimension N is not too large

LINEAR SEPARABLE CASE



- classifiers: $H = \{\mathbf{x} \to \operatorname{sgn}(\mathbf{w}\cdot\mathbf{x}+b), \mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R}\}$
- geometric margin: $\rho = \min_{i\in[1,m]} \frac{|\mathbf{w}\cdot\mathbf{x}_i+b|}{\|\mathbf{w}\|}$

Optimal Hyperplane (V.& C., 1965)



$$\rho = \max_{\mathbf{w},b:\, y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 0} \ \min_{i\in[1,m]} \frac{|\mathbf{w}\cdot\mathbf{x}_i + b|}{\|\mathbf{w}\|}$$

Optimal Hyperplane (V.& C., 1965)

$$\rho = \max_{\mathbf{w}, b\colon y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 0} \min_{i\in[1,m]} \frac{|\mathbf{w}\cdot\mathbf{x}_i + b|}{\|\mathbf{w}\|}$$

$$= \max_{\substack{\mathbf{w}, b\,:\; y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 0 \\ \min_{i\in[1,m]} |\mathbf{w}\cdot\mathbf{x}_i+b| = 1}} \min_{i\in[1,m]} \frac{|\mathbf{w}\cdot\mathbf{x}_i + b|}{\|\mathbf{w}\|} \quad \text{(scale-invariance)}$$

$$= \max_{\substack{\mathbf{w}, b\,:\; y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 0 \\ \min_{i\in[1,m]} |\mathbf{w}\cdot\mathbf{x}_i+b| = 1}} \frac{1}{\|\mathbf{w}\|}$$

$$= \max_{\mathbf{w}, b\colon y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 1} \frac{1}{\|\mathbf{w}\|}$$

Optimization Problem Statement

- **Constrained Optimization**:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1,\ i \in [1, m]$$

- **Properties**:
  — Convex optimization
  — Unique solution for linearly separable case

Optimal Hyperplane

- **Lagrangian**: for all $\mathbf{w}, b, \alpha_i \geq 0$

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1]$$

- **KKT conditions**:

$$\nabla_{\mathbf{w}} L = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = 0 \Leftrightarrow \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

$$\nabla_b L = -\sum_{i=1}^{m} \alpha_i y_i = 0 \Leftrightarrow \sum_{i=1}^{m} \alpha_i y_i = 0$$

$$\forall i \in [1, m],\ \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] = 0$$

Support Vectors

- **Complementary conditions**:

$$\alpha_i[y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] = 0 \Rightarrow \ \alpha_i = 0 \text{ or } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1$$

- **Support vectors**: vectors $\mathbf{x}_i$ such that

$$\alpha_i \neq 0 \ \text{ and } \ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1$$

Dual Optimization Problem (I)

- Plugging optimal $\mathbf{w}$ in L we get

$$L = \underbrace{\frac{1}{2}\left\|\sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i\right\|^2 - \sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)}_{-\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)}$$

$$-\underbrace{\sum_{i=1}^{m}\alpha_i y_i b}_{=0} + \sum_{i=1}^{m}\alpha_i$$

- Thus

$$L = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)$$

Dual Optimization Problem (II)

- **Constrained Optimization**:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

$$\text{s.t. } \alpha_i \geq 0 \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0,\ i \in [1, m]$$

- **Solution**

$$h(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + b \right),$$

with $b = y_i - \sum_{j=1}^{m} \alpha_j y_j (\mathbf{x}_j \cdot \mathbf{x}_i)$ for any SV $\mathbf{x}_i$

Leave-One-Out Error

- **Definition**: let h_S be the hypothesis output by learning algorithm $\mathcal{L}$ after receiving sample S of size m. Then, the LOO error of $\mathcal{L}$ over S is

$$\hat{R}_{loo}(\mathcal{L}) = \frac{1}{m}\sum_{i=1}^{m} 1_{h_{S\setminus\{\mathbf{x}_i\}}(\mathbf{x}_i)\neq f(\mathbf{x}_i)}$$

- **Property**

$$\begin{aligned}\mathbb{E}_{S\sim D^m}[\hat{R}_{loo}(\mathcal{L})] &= \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_S[1_{h_{S\setminus\{\mathbf{x}_i\}}(\mathbf{x}_i)\neq f(\mathbf{x}_i)}] \\ &= \mathbb{E}_S[1_{h_{S\setminus\{\mathbf{x}\}}(\mathbf{x})\neq f(\mathbf{x})}] \\ &= \mathbb{E}_{S'\sim D^{m-1}}[\mathbb{E}_{\mathbf{x}\sim D}[1_{h_{S'}(\mathbf{x})\neq f(\mathbf{x})}]] \\ &= \mathbb{E}_{S'\sim D^{m-1}}[R(h_{S'})]\end{aligned}$$

Leave-One-Out Properties

- **Theorem**: let h_S be the optimal hyperplane for a sample S and let $N_{SV}(S)$ be the number of support vectors defining h_S. Then,

$$\mathbb{E}_{S\sim D^m} R(h_S) \leq \mathbb{E}_{S\sim D^{m+1}} \left[\frac{N_{SV}(S)}{m+1}\right]$$

- **Proof**: let $S \sim D^{m+1}$ be a sample linearly separable and let $\mathbf{x} \in S$. If $h_{S\setminus\{\mathbf{x}\}}$ misclassifies $\mathbf{x}$, then $\mathbf{x}$ must be a SV for h_S. Thus,

$$\hat{R}_{loo}(h^{opt}) \leq \frac{N_{SV}(S)}{m+1}$$

Non-separable case

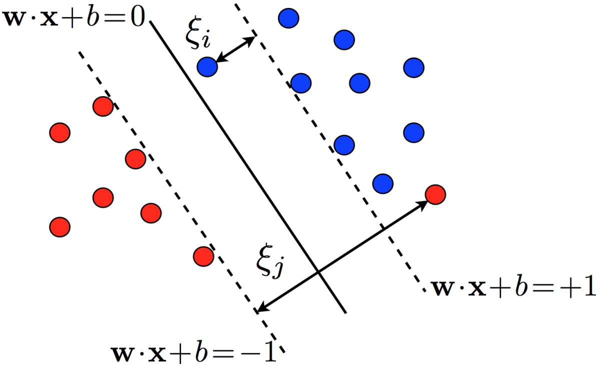
- **Problem**: data often not linearly separable in practice. For any hyperplane there exists $\mathbf{x}_i$, such that

$$y_i[\mathbf{w} \cdot \mathbf{x}_i + b] \ngeq 1$$

- **Approach**: relax constraints using slack variables $\xi_i \geq 0$

$$y_i[\mathbf{w} \cdot \mathbf{x}_i + b] \geq 1 - \xi_i$$

Soft-margin hyperplane



- **Support vectors**: points along the margin or outliers
- **Soft margin**: $\rho = \frac{1}{\|\mathbf{w}\|}$

Optimization Problem Statement

- **Constrained Optimization**:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \xi_i$$

s.t. $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0,\ i \in [1, m]$

- **Properties**:
  — Convex optimization
  — Unique solution
  — $C \geq 0$ is a trade-off parameter

Comments

- How to determine C?
- The problem of determining a hyperplane minimizing the train error is NP-complete (as a function of dimension)
- Other convex functions of the slack variables can be used

Optimal Hyperplane

- **Lagrangian**: for all $\mathbf{w}, b, \alpha_i \geq 0, \beta_i \geq 0$

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{m}\beta_i\xi_i$$

- **KKT conditions**:

$$\nabla_{\mathbf{w}} L = \mathbf{w} - \textstyle\sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i = 0 \Leftrightarrow \mathbf{w} = \textstyle\sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i$$

$$\nabla_b L = -\textstyle\sum_{i=1}^{m}\alpha_i y_i = 0 \Leftrightarrow \textstyle\sum_{i=1}^{m}\alpha_i y_i = 0$$

$$\nabla_{\xi_i} L = C - \alpha_i - \beta_i = 0 \Leftrightarrow \alpha_i + \beta_i = C$$

$$\forall i \in [1, m],\ \alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i] = 0 \text{ and } \beta_i\xi_i = 0$$

Support Vectors

- **Complementary conditions**:

$$\alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1+\xi_i] = 0 \Rightarrow\ \alpha_i = 0 \text{ or } y_i(\mathbf{w}\cdot\mathbf{x}_i+b) = 1-\xi_i$$

- **Support vectors**: vectors $\mathbf{x}_i$ such that

$$\alpha_i \neq 0 \ \text{ and } \ y_i(\mathbf{w}\cdot\mathbf{x}_i+b) = 1-\xi_i$$

Dual Optimization Problem (I)

- Plugging optimal $\mathbf{w}$ in L we get

$$L = \underbrace{\frac{1}{2}\left\|\sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i\right\|^2 - \sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)}_{-\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)}$$

$$-\underbrace{\sum_{i=1}^{m}\alpha_i y_i b}_{=0} + \sum_{i=1}^{m}\alpha_i$$

- Thus

$$L = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)$$

- The condition $\beta_i \geq 0$ is equivalent to $\alpha_i \leq C$

DUAL OPTIMIZATION PROBLEM (II)

- **Constrained Optimization**:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

$$\text{s.t. } 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0,\ i \in [1, m]$$

- **Solution**

$$h(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + b \right),$$

with $b = y_i - \sum_{j=1}^{m} \alpha_j y_j (\mathbf{x}_j \cdot \mathbf{x}_i)$ for any SV $\mathbf{x}_i$ with $0 < \alpha_i < C$