I2ML :: CHEAT SHEET

The **I2ML**: Introduction to Machine Learning course offers an introductory and applied overview of "supervised" Machine Learning. It is organized as a digital lecture.

Classification

Given a **classification problem:**

$$\begin{array}{rl} x \in \mathcal{X} & \text{feature vector} \\ y \in \mathcal{Y} = \{1, \ldots, g\} & \textit{categorical} \text{ output variable (label)} \\ \mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(n)}, y^{(n)}\right)\right) & \text{observations of } x \text{ and } y \end{array}$$

Classification: to construct g discriminant functions: $f_1(\mathbf{x}), \ldots, f_g(\mathbf{x})$, so that we choose our class as $h(\mathbf{x}) = \arg\max_{k \in \{1,\ldots,g\}} f_k(\mathbf{x})$

Linear Classifier

If discriminants $f_k(\mathbf{x})$ can be written as affine linear functions, possibly through a rank-preserving, monotone transformation g:

$$g(f_k(\mathbf{x})) = \boldsymbol{w}_k^\top \mathbf{x} + b_k,$$

we will call the classifier **linear**.

If there exists a linear classifier that perfectly separates the classes of some dataset, the data are called **linearly separable**.

Note: Linear classifiers can represent **non-linear** decision boundaries in the original input space if we use derived features like higher order interactions, polynomial features, etc.

Binary classification

Only 2 classes, can use a single discriminant function $f(\mathbf{x}) = f_1(\mathbf{x}) - f_2(\mathbf{x})$.

Generative Approach

Generative approach models $p(\mathbf{x}|y = k)$, usually by making some assumptions about the structure of these distributions and employs the Bayes theorem:

$\pi_k(\mathbf{x}) = \mathbb{P}(y = k \mid \mathbf{x}) \propto p(\mathbf{x}|y = k)\pi_k$. It allows the computation of $\pi_k(\mathbf{x})$.

Examples:

- Linear discriminant analysis (LDA)
- Quadratic discriminant analysis (QDA)
- Naive Bayes

Linear Discriminant Analysis (LDA): follows a generative approach, each class density is modeled as a *multivariate Gaussian* with equal covariance, i. e. $\Sigma_k = \Sigma \quad \forall k$.

Parameters θ are estimated in a straight-forward manner by estimating $\hat{\pi}_k$, $\hat{\mu}_k$, $\hat{\Sigma}$.

- Each class fit as a Gaussian distribution over the feature space
- Different means but same covariance for all classes
- Rather restrictive model assumption.

Quadratic Discriminant Analysis (QDA): is a direct generalization of LDA, where the class densities are now Gaussians with unequal covariances Σ_k.

Parameters θ are estimated in a straight-forward manner by estimating $\hat{\pi}_k$, $\hat{\mu}_k$, $\hat{\Sigma}_k$.

- Covariance matrices can differ over classes.
- Yields better data fit but also requires estimation of more parameters.

Naive Bayes classifier: A "naive" conditional independence assumption is made: the features given the category y are conditionally independent of each other

- Covariance matrices can differ over both classes but assumed to be diagonal.
- Assumption of uncorrelated features. Often performs well despite this usually wrong assumption.
- Easy to deal with mixed features (metric and categorical)

Discriminant Approach

Discriminant approach: tries to optimize the discriminant functions directly, usually via empirical risk minimization:

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \mathcal{R}_{\text{emp}}(f) = \arg\min_{f \in \mathcal{H}} \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right).$$

Examples:

- Logistic/Softmax regression
- KNN

Logistic Regression

Directly modeling the posterior probabilities $\pi(\mathbf{x})$ of the labels is **logistic regression**.
Encode $y \in \{0, 1\}$ and use ERM. Then model:

$$\pi(\mathbf{x}) = \mathbb{P}(y = 1 \mid \mathbf{x}) = \theta^T x$$

. To avoid predicted probabilities $\pi(\mathbf{x}) \notin [0, 1]$, logistic regression "squashes" the estimated linear scores $\theta^T x$ to $[0, 1]$ through the **logistic function** s:

$$\pi(\mathbf{x}) = \frac{\exp\left(\theta^T x\right)}{1 + \exp\left(\theta^T x\right)} = \frac{1}{1 + \exp\left(-\theta^T x\right)} = s\left(\theta^T x\right).$$

The inverse $s^{-1}(\pi) = \log\left(\frac{\pi}{1-\pi}\right)$ is called **logit** or **log odds**.

Cross Entropy/Bernoulli/Log-loss

Minimizing it refers to maximizing the probabilities of logistic regression.

- $\mathcal{Y} = \{0, 1\}$ with score function: $L(y, f) = -y \cdot f + \log(1 + \exp(f))$
- $\mathcal{Y} = \{-1, +1\}$ with score function: $L(y, f) = \log(1 + \exp(-yf))$
- $\mathcal{Y} = \{0, 1\}$ with probability function:
$L(y, \pi) = -y \log(\pi) - (1 - y) \log(1 - \pi)$
- $\mathcal{Y} = \{-1, +1\}$ with probability function:
$L(y, \pi) = -\frac{1+y}{2} \log(\pi) - \frac{1-y}{2} \log(1 - \pi)$

Softmax

A generalization of the logistic function. It "squashes" a g-dimensional real-valued vector z to a vector of the same dimension, with every entry in the range [0, 1] and all entries adding up to 1.
Softmax is defined on a numerical vector z: $s_k(z) = \frac{\exp(z_k)}{\sum_j \exp(z_j)}$.