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

We want to assign new observations to known categories according to criteria learned from a training set.

Assume we are given a **classification problem:**

$x \in \mathcal{X}$	feature vector
$y \in \mathcal{Y} = \{1, \dots, g\}$	*categorical* output variable (label)
$\mathcal{D} = \left\{ \left(\mathbf{x}^{(1)}, y^{(1)}\right), \dots, \left(\mathbf{x}^{(n)}, y^{(n)}\right) \right\}$	observations of x and y

Classification usually means to construct g discriminant functions:

$f_1(x), \dots f_g(x)$, so that we choose our class as $h(x) = \arg\max_k f_k(x)$ for $k = 1, 2, \dots, g$

Linear Classifier:

If the functions $f_k(x)$ can be specified as linear functions, we will call the classifier a *linear classifier*.

Note: All linear classifiers can represent non-linear decision boundaries in our original input space if we include derived features. For example: higher order interactions, polynomials or other transformations of x in the model.

Binary classification: If only 2 classes exist

We can use a single discriminant function $f(x) = f_1(x) - f_2(x)$.

Generative Approach

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

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

Examples of **Generative** approach:

Generative Approach models $p(x|y)$ and $p(y)$.
Example:

1. Linear discriminant analysis (LDA)
2. Quadratic discriminant analysis (QDA)
3. Naïve 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 $\hat{\pi}_k$, $\hat{\mu}_k$, $\hat{\Sigma}$.

1. Each class fit as a Gaussian distribution over the feature space
2. Different means but same covariance for all classes
3. 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 $\hat{\pi}_k$, $\hat{\mu}_k$, $\hat{\Sigma}_k$.

1. Covariance matrices can differ over classes.
2. 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

1. Covariance matrices can differ over both classes but assumed to be diagonal.
2. Assumption of uncorrelated features. Often performs well despite this usually wrong assumption.
3. 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 of **Discriminant** approach:

Discriminant Approach: models $p(y|x)$ directly.
Example:

1. Logistic/Softmax regression
2. kNN

Logistic Regression: A *discriminant* approach for directly modeling the posterior probabilities $\pi(\mathbf{x})$ of the labels is **logistic regression**

We focus on the binary case $y \in \{0, 1\}$. We then model:

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

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

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

Cross Entropy loss: Minimizing it refers to maximizing the probabilities of logistic regression, where the labels are $\mathcal{Y} = \{0, 1\}$

$L(y, f(\mathbf{x})) = y\theta^T x - \log[1 + \exp(\theta^T x)]$

Bernoulli Loss: If we encode the labels with $\mathcal{Y} = \{-1, +1\}$ we can simplify the loss function as: $L(y, f(\mathbf{x})) = \log[1 + \exp(-yf(\mathbf{x})]$

Bernoulli loss is equivalent to Cross Entropy loss encoded differently.

Softmax: is 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:

$$s(z)_k = \frac{\exp(z_k)}{\sum_j \exp(z_j)}$$