IntroML - Lecture Notes Week 10

Ruben Schenk, ruben.schenk@inf.ethz.ch

July 19, 2022

1 Statistical Perspective on Supervised Learning

1.1 Introduction

We have seen how we can fit prediction models for regression and classification. We will now explore a **statistical perspective** on supervised learning, i.e. estimate the data distribution and derive some prediction/decision rule from the distribution.

Benefits are:

- Quantify the uncertainty
- Express prio knowledge/assumptions about the data
- Allows to derive new methods

Recall the general goal of supervised learning: Given the training data $D = \{(x_1, y_i), ..., (x_n, y_n)\} \subseteq X \times Y$. We want to identify a hypothesis $f : X \to Y$ from some class F, e.g.:

- Linear models: $f(x) = w^T x$
- Kernel methods: $f(x) = \sum_i \alpha_i k(x_i, x)$
- Neural networks: $f(x) \sum_i w_i' \sigma(w_i^T x)$

Our goal is to minimize the prediction error, i.e. the loss on unseen examples. But what does this mean more formally? When can we hope that learning will succeed?

The fundamental assumption is that our data set is generated independently and identically distributed (i.i.d.), i.e.

$$(x_i, y_i) \sim p(x, y)$$

for some unknown p. We would like to identify a hypothesis $f : X \to Y$ that minimizes the expected loss (prediction error, population risk):

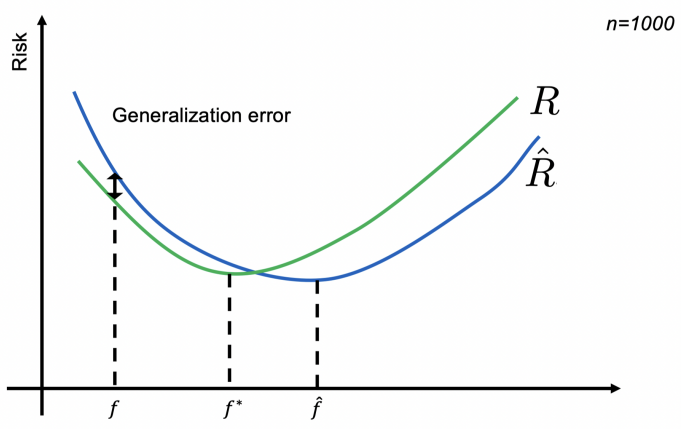
$$R(f) = \int p(x, y) l(y; f(x))\, dxdy = \mathbb{E}_{x,y}[l(y; f(x))].$$

1.2 Estimating Generalized Error

We can estimate the generalized error by estimating the true risk by the empirical risk on the sample data set D (*Empirical Risk Minimization (ERM)*):

$$\hat{R}_D(f) = \frac{1}{|D|} \sum_{(x,y) \in D} l(y; f(x)).$$

Why might this work? Because of the law of large numbers (LLN): $\hat{R}_D(f) \to R(f)$ almost surely as $|D| \to \infty$.



The more samples you take (higher n), the closer the two curves go together!

What happens if we optimize on training data? Assume we are given training data D and obtain some solution $\hat{f}_D = \arg\min_{f \in F} \hat{R}_D(f)$. Ideally, we wish to solve for $f^* = \arg\min_{f \in F} R(f)$. However, in general, it will hold that $\mathbb{E}_D[\hat{R}_D(\hat{f}_D)] \leq \mathbb{E}_D[R(\hat{f}_D)]$. Thus, we obtain an *overly optimistic estimate.*

What would be a more realistic evaluation? We want to avoid underestimating the prediction error. The idea is as follows: We obtain training (D) and test data (D') from the same distribution p. Then we optimize f on the training set, i.e. $\hat{f}_D = \arg\min_{f \in F} \hat{R}_D(f)$. We evaluate on the test set

$$\hat{R}_{D'}(\hat{f}_D) = \frac{1}{|D'|} \sum_{(x,\, y) \in D'} l(y;\, \hat{f}_D(x)).$$

Then it holds that:

$$\mathbb{E}'_D[\hat{R}_{D'}(\hat{f}_D)] = R(\hat{f}_D)$$

The i.i.d. assumption is a standard assumption, but often violated in practice! E.g. temporal dependencies, spatial/geographic dependencies, sampling bias, strategic behavior, etc.

1.3 Optimal Predictor for Squared Loss

For the squared loss, the population risk is

$$R(f) = \mathbb{E}_{x,\, y}[(y - f(x))^2].$$

Suppose we (unrealistically) knew $p(x,\, y)$, and we allow arbitrary functions f. Which f minimizes the risk?

Assuming the data is generated i.i.d. according to $(x_i,\, y_i) \sim p(x,\, y)$. The hypothesis f^* minimizing $R(f)$ is given by the **conditional mean:**

$$f^*(x) = \mathbb{E}[Y \,|\, X = x]$$

This (in practice unattainable) hypothesis is called the **Baye's optimal predictor** for the squared loss. **Note:** We only need the conditional distribution $p(y \,|\, x)$. not the full joint distribution $p(x,\, y)$.

In practice, we have finite data. We know that $f^*(x) = \mathbb{E}[Y \,|\, X = x]$. Thus, one strategy for estimating a predictor from training data is to estimate the conditional distribution $\hat{p}(y \,|\, x)$ and then, for some test point x, predict the label via the conditional mean:

$$\hat{y} = \hat{\mathbb{E}}[Y \,|\, X = x] := \int y \hat{p}(y \,|\, x)\, dy$$

The common approach for estimating conditional distributions is by **parametric estimation** $\hat{p}(y \,|\, x,\, \theta)$:

1. Choose a particular parametric form

2. Then estimate the parameters θ

Step 2 is done using maximum (conditional) Likelihood estimation:

$$\theta^* = \arg\max_{\theta} \hat{p}(y_1, ..., y_n \,|\, x_1, ..., x_n, \theta)$$

1.4 Probabilistic Model for Regression

Consider linear regression. Let's make the statistical assumption that the noise is Gaussian, i.e.:

$$y_i = w^T x_i + \epsilon_i \text{ where } \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Then we can compute the conditional likelihood of the data given any candidate model w as:

$$\hat{w} = \arg\min_{w} \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

thus, under the "conditional linear Gaussian" assumption, maximizing the likelihood is equivalent to least squares estimation.

This observation is actuall not tied to linear models. Suppose $F = \{f : X \to \mathbb{R}\}$ is a class of functions. Assuming that $p(Y = y \,|\, X = x) = \mathcal{N}(y \,|\, f^*(x),\ \sigma^2)$ for some function $f^* : X \to Y$ in F and some $\sigma^2 > 0$, the MLE for data $D = \{(x_1, y_1), ..., (x_n, y_n)\}$ is given by

$$\hat{f} = \arg\min_{f \in F} \sum_{i=1}^{n} (y_i - f(x_i))^2.$$

In summary, the maximum likelihood estimate (MLE) is given by the least squares solution, assuming that the noise is i.i.d. Gaussianm with constant variance. This is useful since MLE satisfies several nice statistical properties (not fromally defined here):

1. Consistency: parameter estimate converges to true parameters in probability
2. Asymptotic efficiency: smallest variance among all "well-behaved" estimators for large n
3. Asymptotic normality

However, all these properties are asymptotic (hold as $n \to \infty$). For finite n, we must avoid overfitting!

1.5 Bias, Variance, and Noise

Recall the **bias variance tradeoff:**

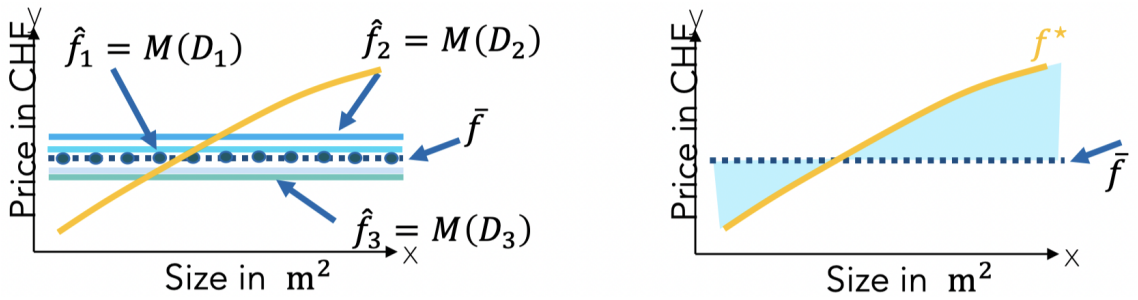
$$\text{Prediction error} = \text{Bias}^2 + \text{Variance} + \text{Noise}$$

- Bias: Excess risk of best model considered compared to minimal achievable risk knowing $p(x, y)$ (i.e. given infinite data)
- Variance: Risk incurred due to estimating model from limited data
- Noise: Risk incurred by optimal model (i.e. irreducible error)

1.5.1 Bias in Estimation

MLE solution depends on training data D. But the training data D is itself random, i.e. drawn i.i.d. from p. We might want to choose F to have small **bias,** i.e. to have a small squared error on average:

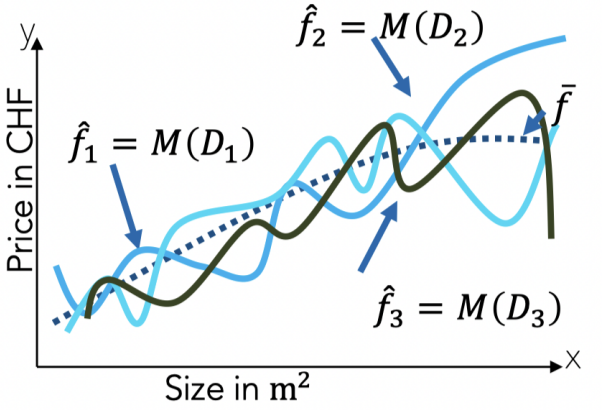
$$\mathbb{E}_x[\bar{f}(x) - f^*(x)]^2 \text{ for } \bar{f}(x) := \mathbb{E}_D[\hat{f}_D(x)].$$



1.5.2 Variance in Estimation

The estimator MLE is itself random, and has some variance:

$$\mathbb{E}_x[\mathrm{Var}_D[\hat{f}_D(x)]] = \mathbb{E}_x[\mathbb{E}_D[(\hat{f}_D(x) - \bar{f}(x))^2]]$$



1.5.3 Noise in Estimation

Even if we know the Bayes' optimal hypothesis f^*, we'd still incur some error due to noise:

$$\mathbb{E}_{x,y}[(y - f^*(x))^2]$$

This error is *irreducible,* i.e. independent of choice of the hypothesis class.

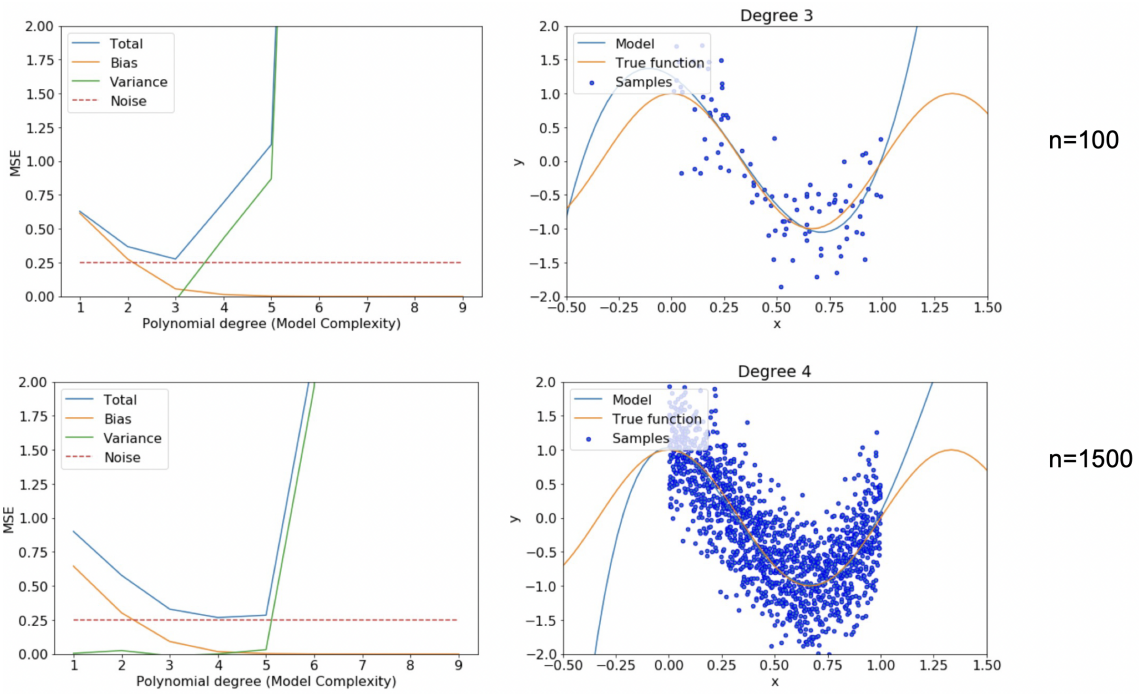
1.5.4 Bias-Variance Tradeoff

For least-squares estimation the following holds:

$$\mathbb{E}_{D,x,y}[(y - \hat{f}_D(x))^2] = \mathbb{E}_x[\bar{f}(x) - f^*(x)]^2 + \mathbb{E}_x[\mathrm{Var}_D[\hat{f}_D(x)]] + \mathbb{E}_{x,y}[y - f^*(x)]^2,$$

$$\text{Expected Mean Squared Prediction Error} = \text{Bias}^2 + \text{Variance} + \text{Noise},$$

where $\hat{f}(x) = \mathbb{E}_D[\hat{f}_D(x)]$ and $\mathrm{Var}_D[\hat{f}_D(x)] = \mathbb{E}_D[(\hat{f}_D(x) - \bar{f}(x))^2]$. Ideally, we wish to find an estimator that simultaneously minimizes bias and variance.



The maximum likelihood estimate for linear regression is unbiased (if $f^* \in F$). Furthermore, it is the maximum variance estimator among all unbiased estimators. However, we have already seen that the least-squares solution can overfit. Thus, we might trade a little bit of bias for a (potentially dramatic) reduction in variance.

1.6 Bayesian Modeling

We can introduce bias by expressing assumptions on parameters through a **Bayesian prior.** For example, let's assume weights are small, more likely around 0. We can capture this assumption with a Gaussian prior $w \sim \mathcal{N}(0, \beta^2 I)$, $w_i \sim \mathcal{N}(0, \beta^2)$. Then, the *posterior distribution* of w is given using **Bayes' rule** by:

$$p(w \mid x_1, ..., x_n, y_1, ..., y_n) = \frac{p(w)p(y_1, ..., y_n \mid x_1, ..., x_n\, w)}{p(y_1, ..., y_n \mid x_1, ..., x_n)}$$

Which parameters w are most likely a posteriori? The solution to this question is given by the *maximum a posteriori estimate:*

$$\hat{w} = \arg\max_w p(w \mid x_1, ..., x_n, y_1, ..., y_n) = \arg\min_w \frac{\sigma^2}{\beta^2}||w||_2^2 + \sum_{i=1}^{n}(y_i - w^T x_i)^2.$$

This is equivalent to the ridge regression solution for $\lambda = \frac{\sigma^2}{\beta^2}$.

Ridge regression can be understood as finding the *Maximum A Posteriori (MAP) parameter estimate* for a linear regression problem, assuming that:

1. The noise $p(y \mid x, w)$ is i.i.d. Gaussian
2. The prior $p(w)$ on the model parameters w is Gaussian

Then:

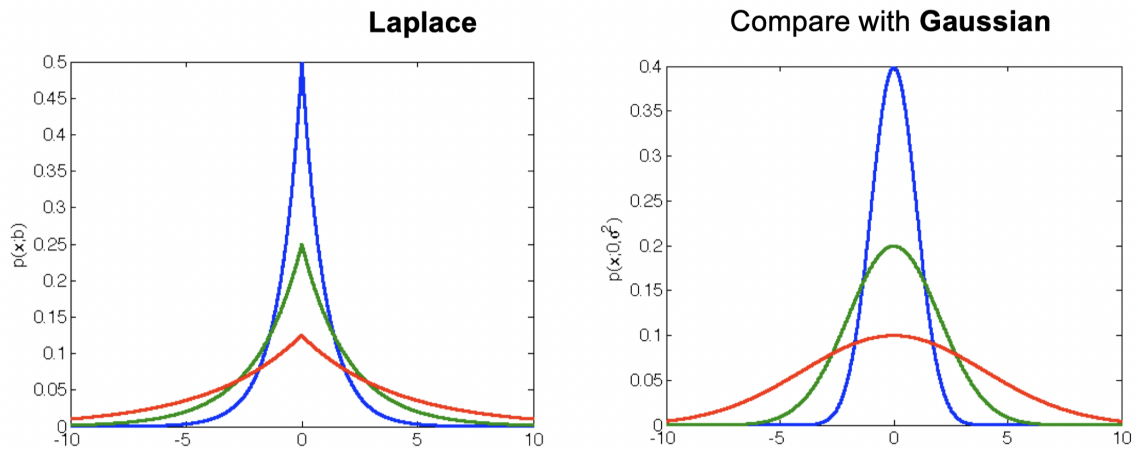
$$\arg\min_w \sum_{i=1}^{n}(y_i - w^T x_i)^2 + \lambda||w||_2^2 \equiv \arg\max_w p(w) \prod_{i=1}^{n} p(y_i \mid x_i, w)$$

More generally, regularized estimation can often be understood as a MAP inference:

$$\arg\min_w \sum_{i=1}^{n} l(w^T x_i;\, x_i,\, y_i) + C(w) \equiv \arg\max_w p(w) \prod_{i=1}^{n} p(y_i \,|\, x_i,\, w) = \arg\max_w p(w \,|\, D),$$

where $C(w) = -\log p(w)$ and $l(w;\, x,\, y) = -\log p(y \,|\, x,\, w)$. This perspective allows changing priors (regularizers) and likelihoods (loss functions).

Example: Is there a prior that corresponds to l1-regularization? Yes, the *Laplace prior.*



$$p(x;\, \mu,\, b) = \frac{1}{2b} \exp\Big(-\frac{|x-\mu|}{b} \Big)$$

1.7 Statistical Models for Classification

In classification (with a 0-1-loss), the population risk is given by:

$$R(f) = P(y \neq f(x)) = \mathbb{E}_{x,\,y}[[y \neq f(x)]].$$

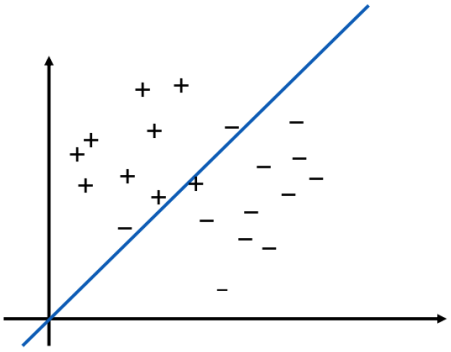
Suppose we (unrealistically) knew $p(X,\, Y)$. Which f minimizes the risk then? Assuming the data is generated i.i.d. according to $(x_i,\, y_i) \sim p(x,\, y)$, then, the hypothesis f^* minimizing $R(f)$ is given by the *most probable class*

$$f^*(x) = \arg\max_y p(Y = y \,|\, X = x).$$

This (in practice unattainable) hypothesis is called the **Bayes' optimal predictor** for the 0-1-loss / misclassification error. Thus, the natural approach is again to estimate $p(y \,|\, x)$.

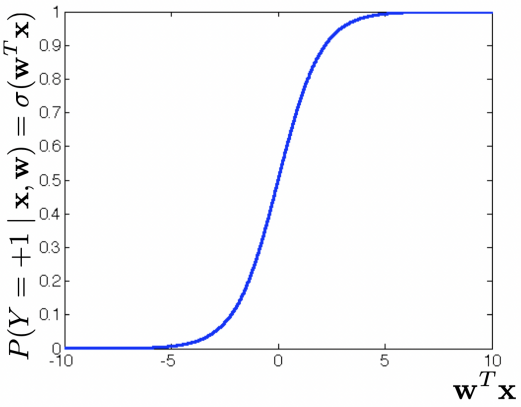
1.8 Logistic Regression

The main idea behind **logistic regression** is to use a generalized linear model for the class probability:



The *logsitc link function* for logistic regression is given by:

$$\sigma(w^T x) = \frac{1}{1 + \exp(-w^T x)}$$



Logistic regression replaces the assumption of Gaussian noise by i.i.d. Bernoulli noise:

$$p(y \mid x, w) = \text{Ber}(y; \sigma(w^T x))$$

But how can we estimate the parameters w? Through maximum likelihood estimation or MAP estimation.

The MLE for logistic regression is the negative log likelihood function given by:

$$L(w) = \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i))$$

The logistic loss is *convex.* Thus, we can use convex optimization techniques, such as SGD.

Similar to SVMs and linear regression we want to use regularizers to control model complexity. Thus, instead of solving the MLE, we estimate MAP / solve the regularized problem:

- L2 (Gaussian prior): $\sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i)) + \lambda ||w||_2^2$
- L1 (Laplace prior): $\sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i)) + \lambda ||w||_1$

The same ideas also apply to **multi-class logistic regression.** We maintain one weight vector per class and model:

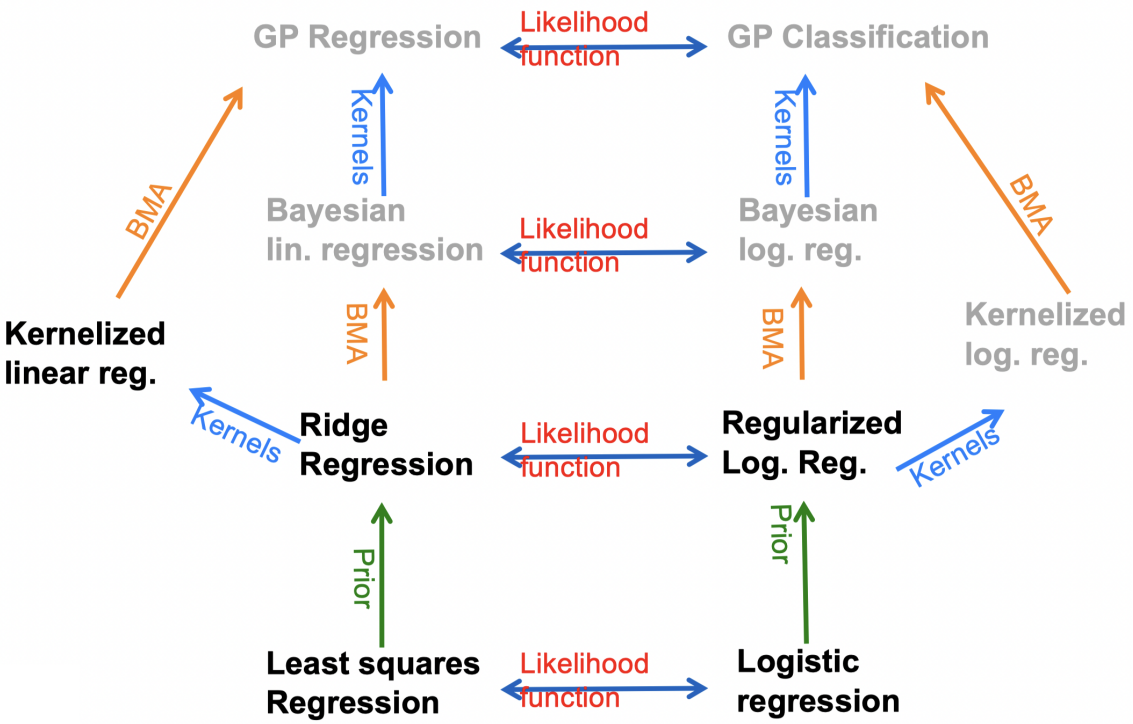
$$p(Y = i \mid x, w_1, ..., w_c) = \frac{\exp(w_i^T x)}{\sum_{j=1}^{c} \exp(w_j^T x)}$$

This is not unique, but we can force uniqueness by setting $w_c = 0$ (this recovers binary logistic regression as special case). The corresponding loss function (also called **cross-entropy loss**) is given by:

$$l(y, x; w_1, ..., w_c) = -\log p(Y = y \mid x, w_1, ..., w_c)$$

1.9 Summary

The summary for both supervised learning and unsupervised learning so far:



Model / function class:	Linear hypotheses; nonlinear hypotheses with nonlinear feature transforms, kernels, learn nonlinear features via neural nets		
Probabilistic interpretation:	Likelihood	*	Prior
Objective:	Loss-function	+	Regularization/penalty
	Squared loss = Gaussian lik., student-t lik. 0/1 loss, logistic loss = Bernoulli lik., cross-entropy loss = categorial lik., Hinge loss, cost sensitive losses, reconstruction error		L^2 norm = Gaussian prior, L^1 norm = Laplace prior, L^0 penalty, early stopping, dropout
Method:	Exact solution (eigendecomposition for PCA), Gradient Descent, (mini-batch) SGD, Reductions, Lloyd's heuristic, Bayesian model averaging		
Evaluation metric:	Mean squared (reconstruction) error, Accuracy, F1 score, AUC, Confusion matrices, log-likelihood on validation set		
Model selection:	K-fold Cross-Validation, Monte Carlo CV, Bayesian model selection		