Lecture 5

Linear classification

Machine Learning

Andrey Filchenkov

13.03.2018

Lecture plan

- Linear classification problem
- Gradient descent
- Heuristics for gradient descent
- Regularization
- Logistic regression

- The presentation is prepared with materials of the K.V. Vorontsov’s course “Machine Leaning”.
- Slides are available online: **goo.gl/Wkif2w**

Lecture plan

- Linear classification problem
- Gradient descent
- Heuristics for gradient descent
- Regularization
- Logistic regression

Problem formulation

Constraint: $Y = \{-1, +1\}$

$T^\ell = \{(x_i, y_i)\}_{i=1}^{\ell}$ is given

Find classifier $a_w(x, T^\ell) = \operatorname{sign}(f(x, w))$.

$f(x, w)$ is a discernment function,

w is a parameter vector.

Key hypothesis: objects are (well-)separable.

Main idea: search among separating surfaces described with $f(x, w) = 0$.

Margin

Margin of object x_i:

$$M_i(w) = y_i f(x_i, w),$$

$M_i(w) < 0$ is an evidence of misclassification.

Margin

Margin of object x_i:

$$M_i(w) = y_i f(x_i, w),$$

$M_i(w) < 0$ is an evidence of misclassification.

We have already defined **margin** of object x_i as

$$M(x_i) = C_{y_i}(x_i) - \max_{y \in Y \setminus \{y_i\}} C_y(x_i),$$

where $C_y(u) = \sum_{i=1}^{\ell} [y(u,i) = y] w(i,u)$, $w(i,u)$ is function of u's ith neighbor importance.

What is their relation?

Empirical risk

Empirical risk:

$$Q\left(a_w, T^\ell\right) = Q(w) = \sum_i^\ell [M_i(w) < 0],$$

it is just the number of errors.

The function is not smooth, so it is hard to find optima.

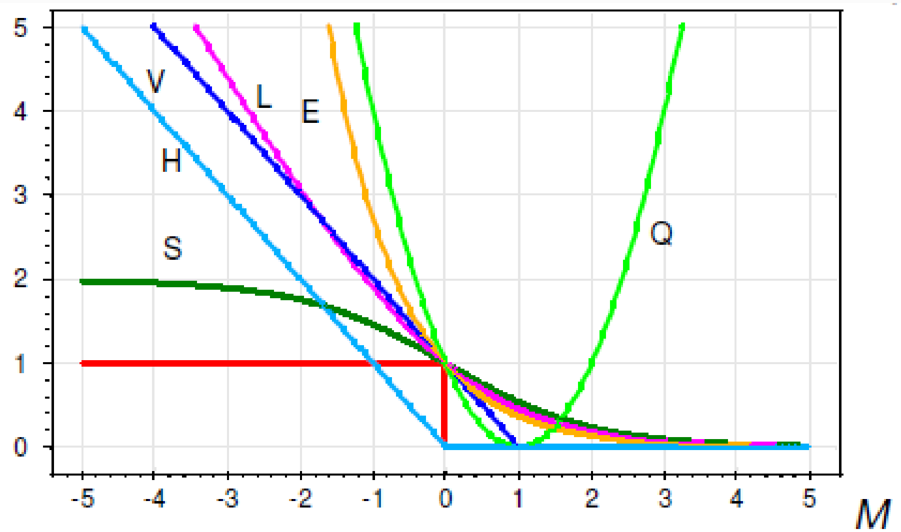
Approximation:

$$\tilde{Q}(w) = \sum_i^\ell L(M_i(w)),$$

where $L(M_i(w)) = L(a_w\left(x_i, T^\ell\right), x_i)$ is a loss function.

Loss function

We want L to be non-negative, non-increasing, and smooth:



$H(M) = (-M)_+$ — piecewise linear (Hebb's rule);

$V(M) = (1 - M)_+$ — piecewise linear (SVM);

$L(M) = \log_2(1 + e^{-M})$ — logarithmic (LR);

$Q(M) = (1 - M)^2$ — square (LDA);

$S(M) = 2(1 + e^{M})^{-1}$ — sigmoid (ANN);

$E(M) = e^{-M}$ — exponential (AdaBoost).

Linear classifier

$f_j\colon X \to \mathbb{R}, j = 1, \dots, n$ are numeric features.

Linear classifier:

$$a_w(x, T^\ell) = \operatorname{sign}\left(\sum_{i=1}^{n} w_i f_i(x) - w_0\right).$$

$w_1, \dots w_n \in \mathbb{R}$ are feature **weights**.

Equivalent notation:

$$a_w(x, T^\ell) = \operatorname{sign}(\langle w, x \rangle),$$

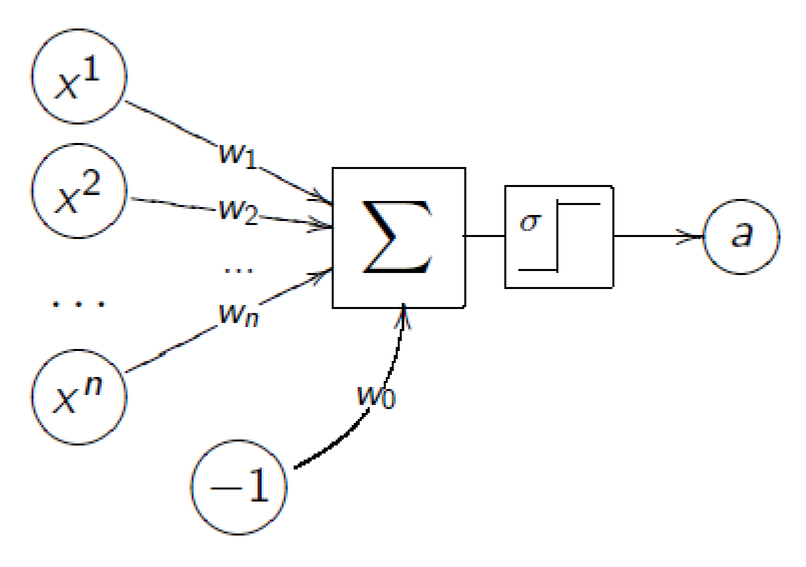
if a feature $f_0(x) = -1$ is added.

Neuron

McCulloch-Pitts neuron:

$$a_w(x, T^\ell) = \sigma\left(\sum_{i=1}^{n} w_i f_i(x) - w_0\right),$$

where σ is an activation function.



Lecture plan

- Linear classification problem
- Gradient descent
- Heuristics for gradient descent
- Regularization
- Logistic regression

Gradient descent

Empirical risk minimization problem

$$\tilde{Q}(w) = \sum_{i}^{\ell} L(M_i(w)) = \sum_{i}^{\ell} L(\langle w, x_i \rangle y_i) \to \min_w.$$

Gradient descent:

$w^{[0]} =$ **an initial guess value**;

$w^{[k+1]} = w^{[k]} - \mu \nabla Q\left(w^{[k]}\right)$,

where μ is a **gradient step**.

$w^{[k+1]} = w^{[k]} - \mu \sum_{i}^{\ell} L'(\langle w, x_i \rangle y_i)\, x_i y_i.$

Stochastic gradient descent

Problem is that there are too many objects, which should be estimated on each step.

Stochastic gradient descent:

$w^{[0]}$ is **an initial guess values**;

$x_{(1)}, \dots, x_{(\ell)}$ is **an objects order**;

$w^{[k+1]} = w^{[k]} - \mu L'\big(\langle w^{[k]}, x_{(k)}\rangle y_{(k)}\big) x_{(k)} y_{(k)}$,

$Q^{[k+1]} = (1-\alpha)Q^{[k]} + \alpha L\big(\langle w^{[k]}, x_{(k)}\rangle y_{(k)}\big)$.

Stop when values of Q and/or w do not change much.

Hebb's rule

Important special case

$$L(a_w, x) = (-\langle w, x\rangle y)_+,$$

where $(s)_+ = s \cdot [s < 0]$.

Hebb's rule (delta rule):

gradient descent step is

if $-\langle w^{[k]}, x_i\rangle y_i > 0$, then $w^{[k]} = w^{[k]} + \mu x_i y_i$.

Rosenblatt perceptron:

$$w^{[k]} = w^{[k]} + \mu(\mathrm{sign}(\langle w, x_i\rangle) - y_i)x_i$$

(the same, when $Y = \{0,1\}$).

Novikov's theorem

Theorem (Novikov)

Let sample T^ℓ be linearly separable: $\exists \widetilde{w}, \exists \delta > 0$:

$\langle \widetilde{w}, x_i \rangle y_i > \delta$ for all $i = 1, \ldots, \ell$.

Them the stochastic gradient descent with Hebb's rule will find weight vector w, which:

- splits sample without error;
- with any initial guess $w^{[0]}$;
- with any learning rate $\mu > 0$;
- independently on objects ordering $x_{(i)}$;
- with finite numbers of changing vector w;
- if $w^{[0]} = 0$, then the number of changes in vector w is

$t_{\max} \leq \frac{1}{\delta^2} \max ||x_j||.$

Lecture plan

- Linear classification problem
- Gradient descent
- Heuristics for gradient descent
- Regularization
- Logistic regression

Heuristics for initial guesses

- $w_j = 0$ for all $j = 0, \ldots, n$;
- small random values: $w_j \in \left[-\frac{1}{2n}, \frac{1}{2n}\right]$;
- $w_j = \frac{\langle y, f_j \rangle}{\langle f_j, f_j \rangle}$;
- learn it with a small random subsample;
- multiply runs with different initial guesses.

Heuristics for object ordering

- take objects from different classes by turns;
- take misclassified objects more frequently;
- do not take "good" object, such that $M_i > \kappa_+$;
- do not take noisy objects, such that $M_i < \kappa_-$.

Heuristics for gradient descent step

- Convergence is achieved for convex functions when
$$\mu^{[k]} \to 0, \sum \mu^{[k]} = \infty, \sum \left(\mu^{[k]}\right)^2 < \infty.$$
- **Steepest gradient descent**:
$$Q\left(w^{[k]} - \mu^{[k]} \nabla Q\left(w^{[k]}\right)\right) \to \min_{\mu^{[k]}}.$$
- Steps for "jog of" local minima.

SG algorithm discussion

Advantages:

- it is easy to implement;
- it is easy to generalize for any f and L;
- dynamical learning;
- can handle small samples.

Disadvantages:

- slow convergence or even divergence is possible;
- can stuck in local minima;
- proper heuristic choice is very important;
- overfitting.

Regularization

Key hypothesis: w "swings" during overfitting

Main idea: clip w norm.

Add regularization penalty for weights norm:

$$Q_\tau(a_w, T^\ell) = Q(a_w, T^\ell) + \frac{\tau}{2}\|w\|^2 \to \min_w.$$

For gradient:

$$\nabla Q_\tau(w) = \nabla Q(w) + \tau w,$$
$$w^{[k+1]} = w^{[k]}(1 - \mu\tau) - \mu\nabla Q(w).$$

Lecture plan

- Linear classification problem
- Gradient descent
- Heuristics for gradient descent
- Regularization
- Logistic regression

Regularization for regression

Key hypothesis: w "swings" during overfitting. This is because of multicollinearity which arises between different features with the growth of the number of features.

Main idea: clip w norm.

Add regularization penalty for weights norm:

$$Q_\tau(a_w, T^\ell) = Q(a_w, T^\ell) + \frac{\tau}{2}||w||^2 \to \min_w.$$

23

Regularization examples

For linear models $A = \{a(x) = \langle w, x \rangle\}$ (regression) and $A = \{a(x) = \mathrm{sign}\langle w, x \rangle\}$ (classification).

L_2-regularization (ridge regression, weight decay):

$$\mathrm{penalty}(A) = \tau \|w\|_2^2 = \tau \sum w_i^2.$$

L_1−regularization (LASSO):

$$\mathrm{penalty}(A) = \tau \|w\|_1 = \tau \sum |w_i|.$$

L_0-regularization (AIC, BIC):

$$\mathrm{penalty}(A) = \tau \|w\|_o = \tau \sum [w_i \neq 0].$$

24

Ridge regression

$$Q\left(a_w, T^\ell\right) + \frac{1}{2\sigma}||w||^2 \rightarrow \min_w$$

Based on idea to clip off “size” of variables.

25

Quadratic penalty conditions

Let $w \in \mathbb{R}^n$ is described with n-dimensional Gaussian distribution:

$$p(w; \sigma) = \frac{1}{(2\pi\sigma)^{n/2}} \exp\left(-\frac{\|w\|^2}{2\sigma}\right),$$

(weights are independent, their expectations are equal to zeros, their variances are the same and equal to σ).

It leads to quadratic penalty:

$$-\ln p(w; \sigma) = \frac{1}{2\sigma}\|w\|^2 + \text{const}(w).$$

LASSO regression

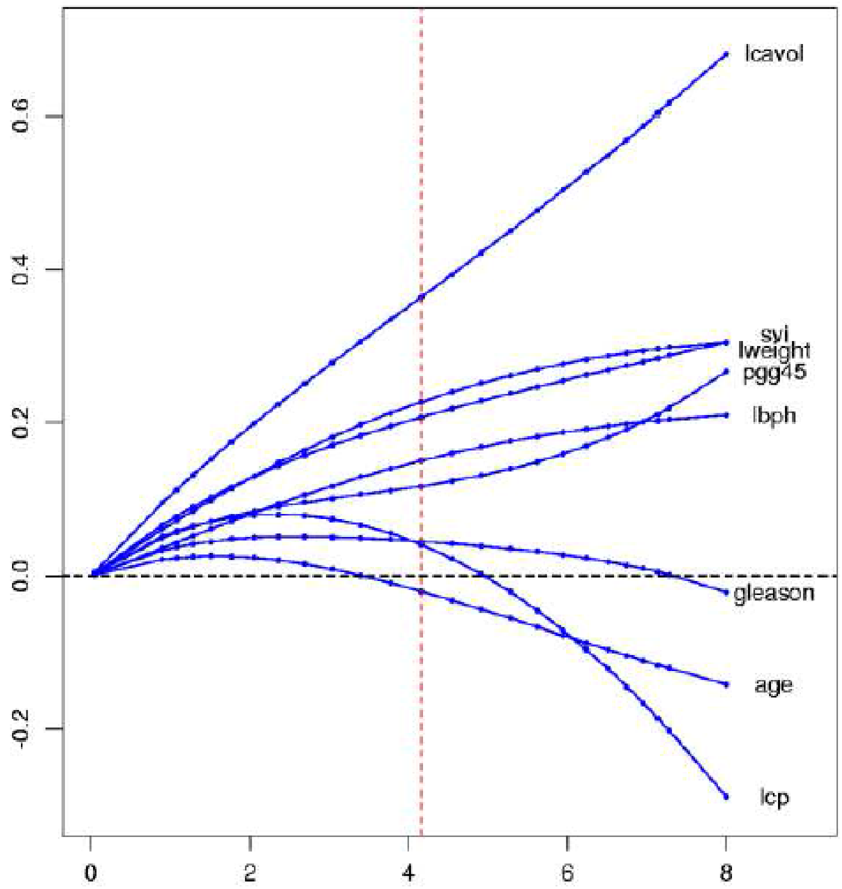
$$Q(a_w, T^\ell) + \kappa|w| \to \min_w$$

Based on idea to clip off the number of variables.

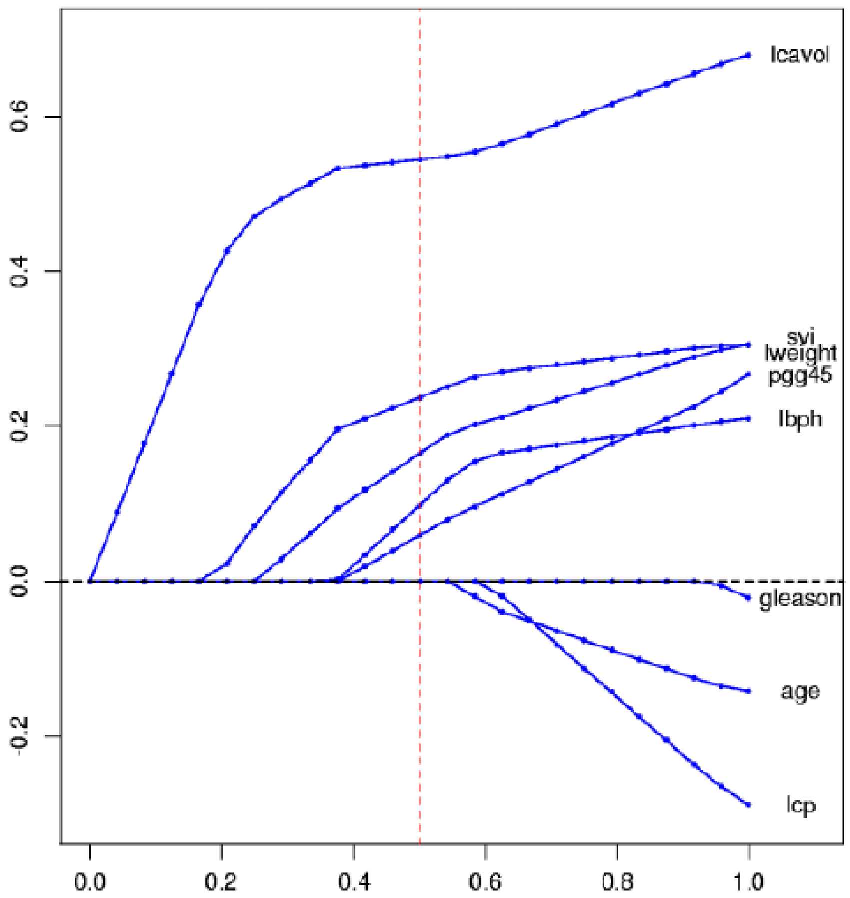
27

Comparison

Ridge regression



Lasso



Regularization for SGD

Add regularization penalty for weights norm:

$$Q_\tau(a_w, T^\ell) = Q(a_w, T^\ell) + \frac{\tau}{2}||w||^2 \to \min_w.$$

For gradient:

$$\nabla Q_\tau(w) = \nabla Q(w) + \tau w,$$
$$w^{[k+1]} = w^{[k]}(1 - \mu\tau) - \mu \nabla Q(w).$$

29

Lecture plan

- Linear classification problem
- Gradient descent
- Heuristics for gradient descent
- Regularization
- Logistic regression

Logistic regression

We may want to talk about probably of belonging to a class (we will discuss it on Lecture 5 in details).

$$y_i = \frac{1}{1 + e^{-\langle w, x_i \rangle}} = \sigma(\langle w, x_i \rangle),$$

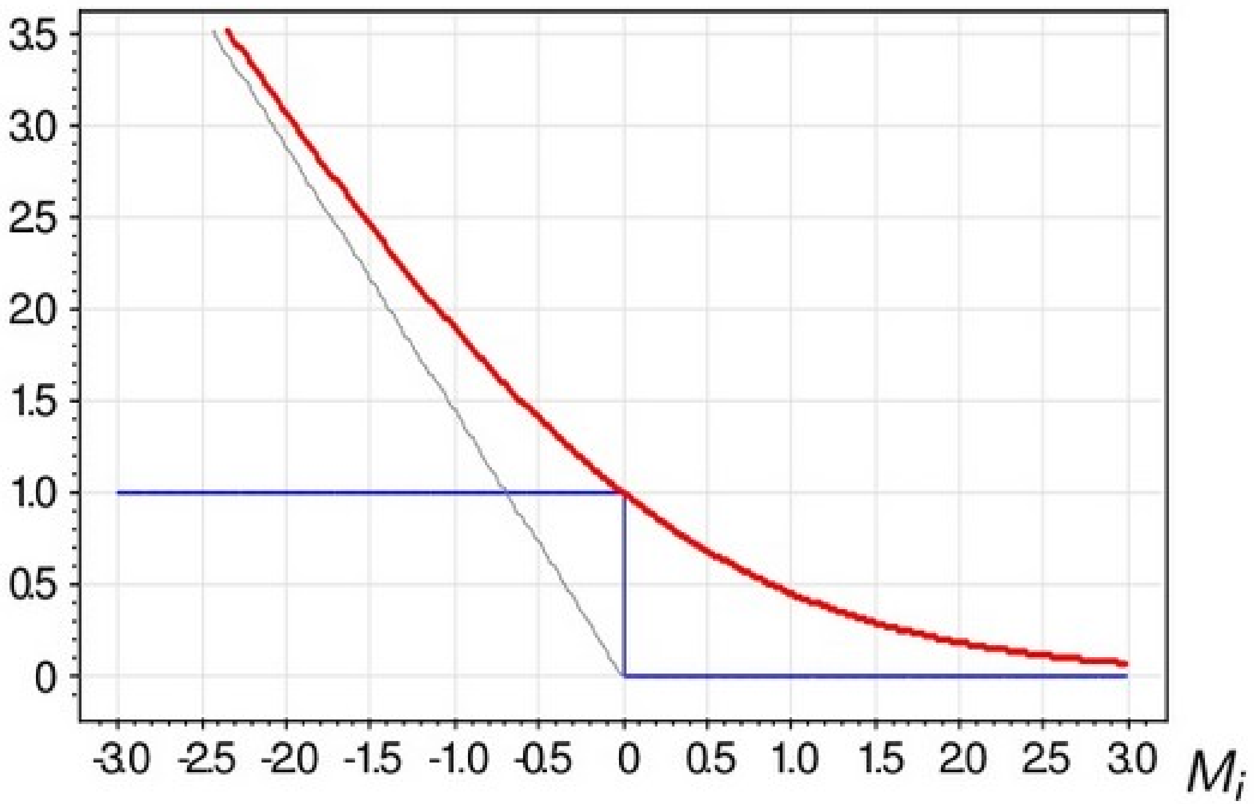
where $\sigma(z)$ is **logistic (sigmoid) function**.

Then classification model is

$$\widetilde{Q_w}\left(a, T^{\ell}\right) = \sum_{i}^{\ell} \ln(1 + \exp(-\langle w, x \rangle y)) \rightarrow \min_w .$$

That is **logarithmic loss function**.

Logarithmic loss function plot



Gradient descent

Derivative:

$$\sigma'(s) = \sigma(s)\sigma(-s).$$

Gradient:

$$\mu\nabla\tilde{Q}\left(w^{[k]}\right) = -\sum_{i}^{\ell} y_i x_i \sigma\left(-M_i(w)\right).$$

Gradient descent step:

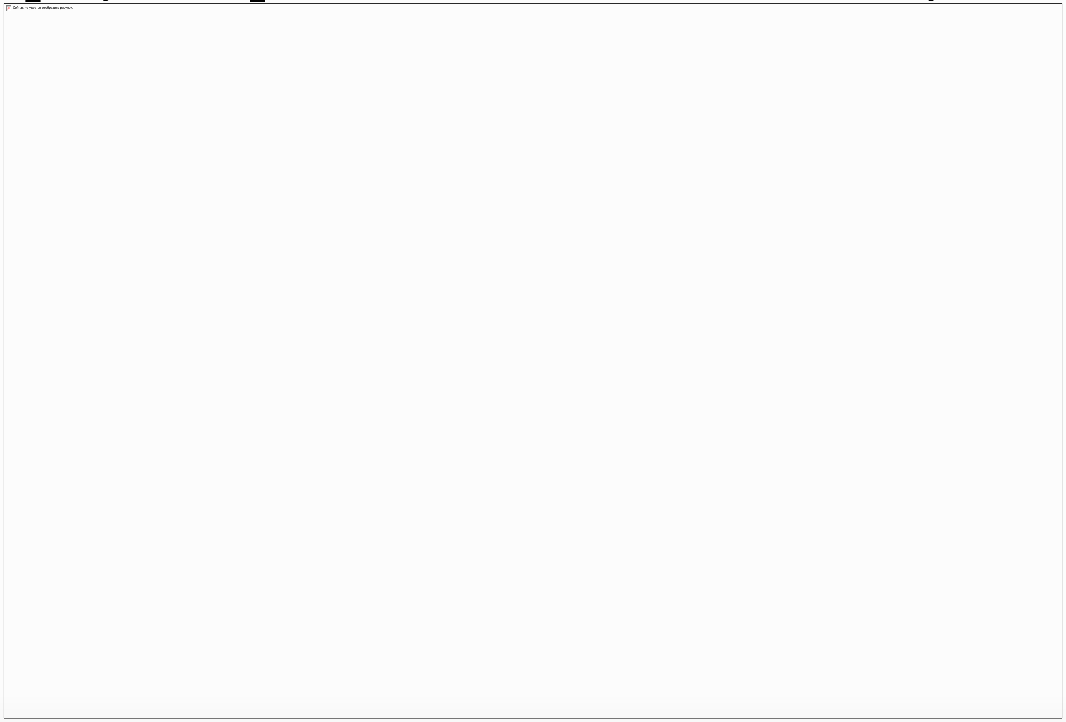
$$w^{[k+1]} = w^{[k]} - \mu y_i x_i \sigma\left(-M_i\left(w^{[k]}\right)\right).$$

Smoothed Hebb's rule

Hebb's rule:

if $-\langle w^{[k]}, x_i \rangle y_i > 0$, then $w^{[k]} = w^{[k]} + \mu x_i y_i$.

Marginal $[M_i < 0]$ and smoothed $\sigma(-M_i)$:



Combining two worlds

Bayesian classifiers

A distribution $p(x, y)$ on object-answers space.

Simple sample of size ℓ $T^{\ell} = \{(x_i, y_i)\}_{i=1}^{\ell}$.

Bayesian classifier: $a_{OB}(x) = \mathrm{argmax}_{y \in Y} \lambda_y \Pr(y)\, p(x|y)$, where λ_y is losses for class y.

Linear classifiers

Constraint: $Y = \{-1, +1\} = \{y_{-1}, y_{+1}\}$

Linear classifier: $a_w(x, T^{\ell}) = \mathrm{sign}(\sum_{i=1}^{n} w_i f_i(x) - w_0)$, where $w_1, \dots, w_n \in \mathbb{R}$ are features weights.

What is their intersection?

Linear Bayesian classifiers

$$Q(a_\theta, T^\ell) = \frac{1}{\ell}\sum\nolimits_1^\ell L(a_\theta, x) = -\sum_{i=1}^{\ell} \ln \varphi(x_i, y_i, \theta) \to \min_\theta .$$

Bayesian classifier for two classes:

$$a(x) = \operatorname{sign}(\lambda_+ \Pr(y_+|x) - \lambda_- \Pr(y_- \,|x)) =$$
$$= \operatorname{sign}\left(\frac{p(x|y_+)}{p(x|y_-)} - \frac{\lambda_- \Pr(y_-)}{\lambda_+ \Pr(y_+)}\right).$$

Separating surface

$$\lambda_+ \Pr(y_+) p(x|y_+) = \lambda_- \Pr(y_-) p(x|y_-)$$

is linear.

Key hypothesis

Key hypothesis: classes are defined with n-dimensional overdispersed exponential densities:

$$p(x|y) = \exp\left(c_y(\delta)\langle\theta_y, x\rangle + b_y\left(\delta, \theta_y\right) + d(x, \delta)\right),$$

where $\theta_y \in \mathbb{R}^m$ is **shift** parameter,

δ is **dispersion** parameter**;**

b_y, c_y, d are some numeric functions.

Overdispersed exponential distribution family includes:

uniform, normal, hypergeometric, Poisson, binominal, Γ-distribution and other.

Example: Gaussian

Let $\theta = \Sigma^{-1}\mu$; $\delta = \Sigma$.

Then

$$\mathcal{N}(x;\mu,\Sigma) = \frac{e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)}}{\sqrt{(2\pi)^n \det \Sigma}} =$$

$$= \exp\left(\left(\mu^\top \Sigma^{-1} x\right) - \left(\frac{1}{2}\mu^\top \Sigma^{-1}\Sigma\Sigma^{-1}\mu\right) - \left(\frac{1}{2}x^\top \Sigma^{-1} x + \frac{n}{2}\ln 2\pi + \frac{1}{2}\ln|\Sigma|\right)\right).$$

The main theorem

Theorem:

If p_y are overdispersed exponential distributions and $f_0(x) = \text{const}$, then

1) Bayesian classifier

$$a(x) = \text{sign}\left(\frac{p(x|y_+)}{p(x|y_-)} - \frac{\lambda_- \Pr(y_-)}{\lambda_+ \Pr(y_+)}\right)$$

is linear: $a(x) = \text{sign}(\langle w, x\rangle - w_0),\ w_0 = \ln\frac{\lambda_-}{\lambda_+}$;

2) posterior probabilities of classes are:

$$\Pr(y|x) = \sigma(\langle w, x\rangle y),$$

where $\sigma(s) = \frac{1}{1+e^{-s}}$, which is **logistic (sigmoid) function**.

Logarithmic loss function

$$\widetilde{Q_w}(a, T^\ell) = \sum_i^\ell L(a, x_i) = \sum_i^\ell \ln p(x_i, y_i; w)$$

$$p(x, y; w) = \Pr(y|x)p(x) = \sigma(\langle w, x\rangle y)\mathrm{const}(w)$$

$$\widetilde{Q_w}(a, T^\ell) = \sum_i^\ell \ln(1 + \exp(-\langle w, x\rangle y)) \to \min_w.$$

That is logarithmic loss function.