Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 1 of 54

Go Back

Full Screen

Close

Quit

1. Klasifikavimo su mokytoju metodai

Klasifikacijos uždavinys yra atpažinimo uždavinys, kurio esmė pagal pateiktus objekto (vaizdo, garso, asmens, proceso) skaitinius duomenis priskirti jį kokiai nors klasei. Laikysime, kad objektas yra aprašomas D-mačiu vektoriumi

$$\mathbf{S} = [s_1, s_2, ..., s_D]',$$

o $\mathbf{Y}_1$, $\mathbf{Y}_2$, ... , $\mathbf{Y}_L$ žymi visas klases, kurioms gali priklausyti objektas. [1] Atliekant klasifikavim1 su mokytoju (kitaip tariant *atpažinimą*), "mokytojas" duoda aibę porų $\{\mathbf{S}_i, y_i\}$, kur $\mathbf{S}_i$ yra i-asis mokymo požymių vektorius, o y_i - i-oji žymė, pažyminti kuriai klasei priklauso požymių vektorius. Pavyzdžiui mokymo duomenimis gali būti vienodo dydžio segmentuotų dešimtainių skaitmenų paveikslėliai,

[1] Čia ir toliau $'$ žymi vektoriaus (arba matricos) transponavimo operaciją

o žymė - dešimtainis skaitmuo, kuris pavaizduotas paveikslėlyje.

Klasifikacijos uždavinys yra sukurti naujai pateikiamų duomenų žymėjimo algoritmą, kuris gerai žymėtų ir pateiktus ir naujai pateikiamus duomenis. Matematine prasme *klasifikatoriumi* vadinamas bet koks $\mathbf{S}$ vektorių atvaizdis į L žymių (kategorijų) aibę $\mathbf{Y}$.

Klasifikavimo uždavinys yra skaidomas į dvi dalis. Pirma yra surandamos objekto savybės $\mathbf{X}$ (angl. *features*). Pažymėkime objekto savybių skaičių n. Savybės yra apskaičiuojamos remiantis pradiniu objektą aprašančiu vektoriumi $\mathbf{S}$. Parenkant objekto savybes yra ieškoma kompromiso tarp

- Mažo savybių skaičiaus n ($n << D$).
- Kuo didesnio savybių informatyvumo. Idealiu atveju iš savybių yra galima rekonstruoti pradinius objekto duomenis $\mathbf{S}$.

Antrajame etape remiantis požymių vektoriumi

$$\mathbf{X} = [x_1, x_2, ..., x_n]'$$

yra atliekama klasifikacija.

Tarkime $\mathbf{S} = [s_1, s_2, ..., s_D]'$ žymi stacionaraus akustinio signalo garso slėgio reikšmes. Tuomet geru požymių rinkiniu yra $\mathbf{S}$ duomenų tiesinės prognozės koeficientai $\mathbf{X} = [x_1, x_2, ..., x_n]'$.

Toliau paprastumo dėlei laikysime, kad yra tik dvi kategorijos, t.y., $L = 2$. Tokios rūšies klasifikavimo uždavinys vadinamas *binariniu*. Nors, atrodytų, daugelio klasių atpažinimo uždavinį galima suskaidyti į binarinius porų klasifikavimo uždavinius, tačiau neakivaizdu kaip apjungti gautus binarinius klasifikatorius į vieną. Daugelio klasių klasifikavimo uždaviniai yra mažiau ištirti ir dažniausiai remiasi "kiekvienas su likusiais" arba "kiekvienas su kiekvienu" strategijomis. Pirmu atveju sudaroma L binarinių klasifikatorių, kurie stengiasi atskirti vieną klasę nuo likusių ir galutinis sprendimas priimamas remiantis L

reikšmėmis gautomis pritaikant binarinius klasifikatorius. Antru atveju sudaroma $L \times (L-1)/2$ binarinių klasifikatorių, kurie stengiasi atskirti visas galimas skirtingų klasių poras ir galutinis sprendimas priimamas remiantis gautomis $L \times (L-1)/2$ reikšmėmis.

Kad sukurti klasifikatorių, reikia turėti *mokymo duomenų*. Pradžioje yra pasirenkama aibė duomenų vektorių S su žinomomis kategorijų/klasių reikšmėmis. Tarkime sudarėme požymių sistemą ir sukūrėme pasirinktai požymių sistemas klasifikatorių. Tuomet natūraliai iškyla klausimas - kokia mūsų klasifikavimo algoritmo kokybė? Šiuo tikslu turima mokymo duomenų aibė skaidoma į tris dalis:

1. mokymo dalis (angl. *training set*)
2. verifikacijos dalis (angl. *validation set*)
3. testavimo dalis. (angl. *test set*).

Yra naudojamos įvairios strategijos kuriant ir įvertinant klasifikavimo algoritmus. Dažnai mokymo ir verifikacijos imtys yra keičiamos ir vidurkinama verifikacijai skirta klasifikavimo statistika. Bene populiariausias yra "kryžminės verifikacijos" (angl. *cross-validation*) metodas. Pagal šį metodą mokymui ir verifikacijai skirti duomenys suskaidomi į apylyges K dalių ir pakaitomis viena iš K dalių laikoma verifikacijos dalimi, o likusios $(K-1)$ skiriamos mokymui. Tokiu būdu galima suvidurkinti K gautų kokybės įverčių ir gauti patikimesnius klasifikavimo kokybės įverčio rezultatus. Taip pat verifikacija naudinga parenkant optimaliai klasifikatoriaus parametrus. Jei klasifikavimo algoritmas linkęs persimokyti, t.y per daug prisitaikyti prie mokymo duomenų specifinių detalių, verifikavimo duomenys padeda aptikti persimokymo momentą ir toliau klasifikavimo algoritmo parametrai nėra tikslinami. Testavimo duomenys skirti įvertinti galutinai klasifikavimo algoritmo kokybę ir yra naudojami *tik vieną*

kartą. Patikrinus klasifikatorių su testavimo duomenimis, klasifikavimo algoritmas toliau yra netobulinamas, nes priešingu atveju vėl galima susidurti su algoritmo "permokymo" problema.

1.1. Tiesiniai klasifikavimo metodai

Dažnai laikoma, kad požymiai $\mathbf{X}$ yra pasiskirstę pagal kokį nors daugiamatį skirstinį. Tokia matematinė abstrakcija suteikia galimybę vienareikšmiai įvertinti klasifikavimo algoritmą kokybę remiantis turima arba įvertinta informacija apie modelio parametrus. Praktiškai visi statistiniai modeliai naudoja kategorijų požymių vidurkius

$$\overline{\mathbf{X}}_1, \overline{\mathbf{X}}_2, ..., \overline{\mathbf{X}}_L.$$

Konkretaus požymių vektoriaus $\overline{\mathbf{X}}$ euklidinio atstumo iki kategorijų vidurkių kvadratas apskaičiuojamas pagal formulę

$$||X - \overline{\mathbf{X}}_l||^2 = (X - \overline{\mathbf{X}}_l)'(X - \overline{\mathbf{X}}_l),$$

Šis atstumas yra vienas galimų panašumo į kategorijas matų. Tokių atstumų pagrindu atlikta klasifikacija vadinama *euklidinio atstumo klasifikatoriumi* (angl. *Euclidean distance clasifier EDC*). Kartais EDC dar vadinami *artimiausiojo vidurkio klasifikatoriumi*.

Dvi pirmąsias kategorijas skirianti riba randama iš lygybės

$$||X - \overline{\mathbf{X}}_1||^2 - ||X - \overline{\mathbf{X}}_l||^2 = 0.$$

Nesunku pastebėti, kad ši riba $\mathbf{X}$ atžvilgiu yra tiesinė. Taigi arčiausiojo vidurkio dviejų kategorijų klasifikatorius apibrėžiamas tiesine *diskriminantine* funkcija

$$h(\mathbf{X}) = \mathbf{X}'\mathbf{V} + c,$$

kur

$$\mathbf{V} = \overline{\mathbf{X}}_1 - \overline{\mathbf{X}}_2$$

ir

$$c = -\frac{\overline{\mathbf{X}}_1 + \overline{\mathbf{X}}_2}{2}.$$

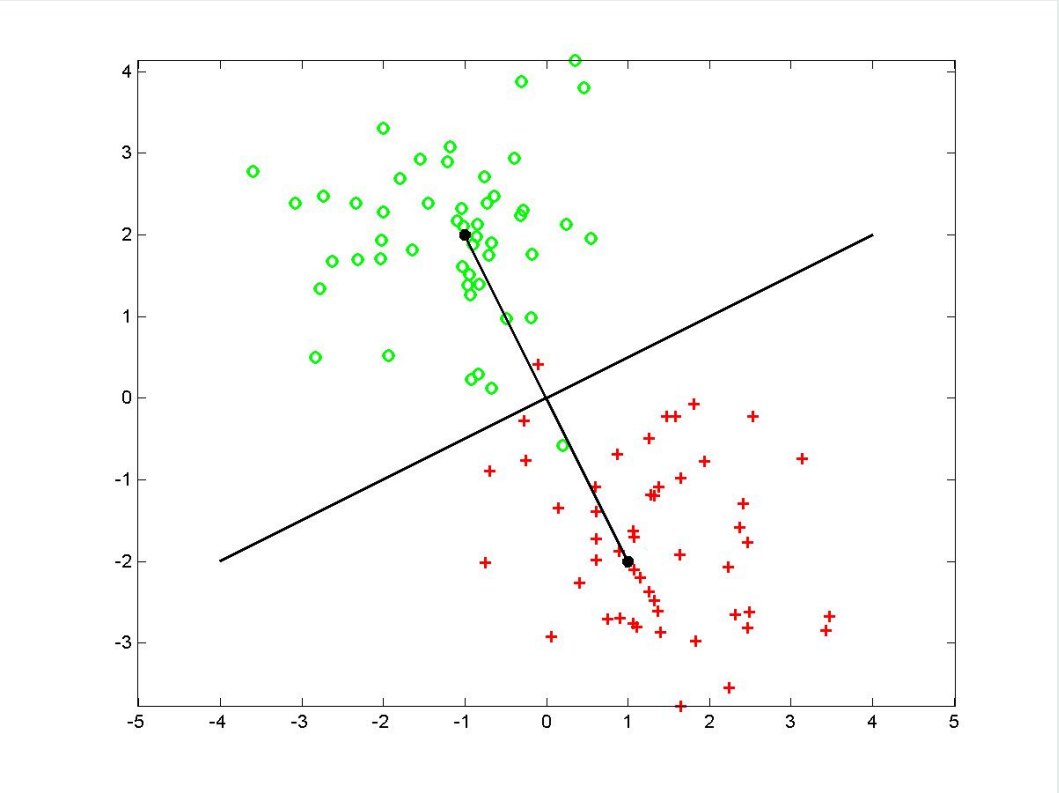
Klasifikavimas atliekamas pagal diskriminantinės funkcijos ženklą - jei jis yra '+', tai $\mathbf{X}$ požymius atitinkantis vektorius priskiriamas 1-ai klasei, priešingu atveju - 2-ai klasei. Žemiau pateikta iliustracija dviejų grupių klasifikavimas naudojant artimiausiojo vidurkio metodą.

Paprastas euklidinio atstumo pagrindu sukurtas klasifikatorius yra asimptotiškai (kai apmokymo duomenų skaičius artėja į begalybę) optimalus, jei duomenys pasiskirstę pagal sferinį Gauso dėsnį

$$N(\mu_l, \sigma^2\mathbf{I}).$$

Čia I žymi vienetinę matricą, o $\sigma^2\mathbf{I}$ - kovariacijos matricą.

Jei požymių kovariacijos matrica Σ nėra įstrižainė, klasifikavimas pagal artimiausiąjį kategorijų vidurkį nėra optimalus. Šiuo at-



1 paveikslėlis: Dviejų grupių artimiausiojo vidurkio klasifikatorius. Grupių centrai: $\overline{\mathbf{X}}_1 = (1, -2)$ ir $\overline{\mathbf{X}}_2 = (-1, 2)$

veju geriau tinka 1936 m. pasiūlytas Fišerio klasifikatorius.

Žemiau pateiktame pavyzdyje palygintos klasifikavimo tiesės gautos taikant Euklidinį atstumą (E) ir Fišerio (F) metodą. Naudojant klasifikavimui pateiktus N_1 ir N_2 duomenis, tiesinio klasifikavimo taisyklė $h(\mathbf{X}) = \mathbf{X}'\mathbf{V}^F + c$ gaunama tokiu būdu:

$$\mathbf{V}^F = \Sigma^{-1}(\overline{\mathbf{X}}_1 - \overline{\mathbf{X}}_2)$$

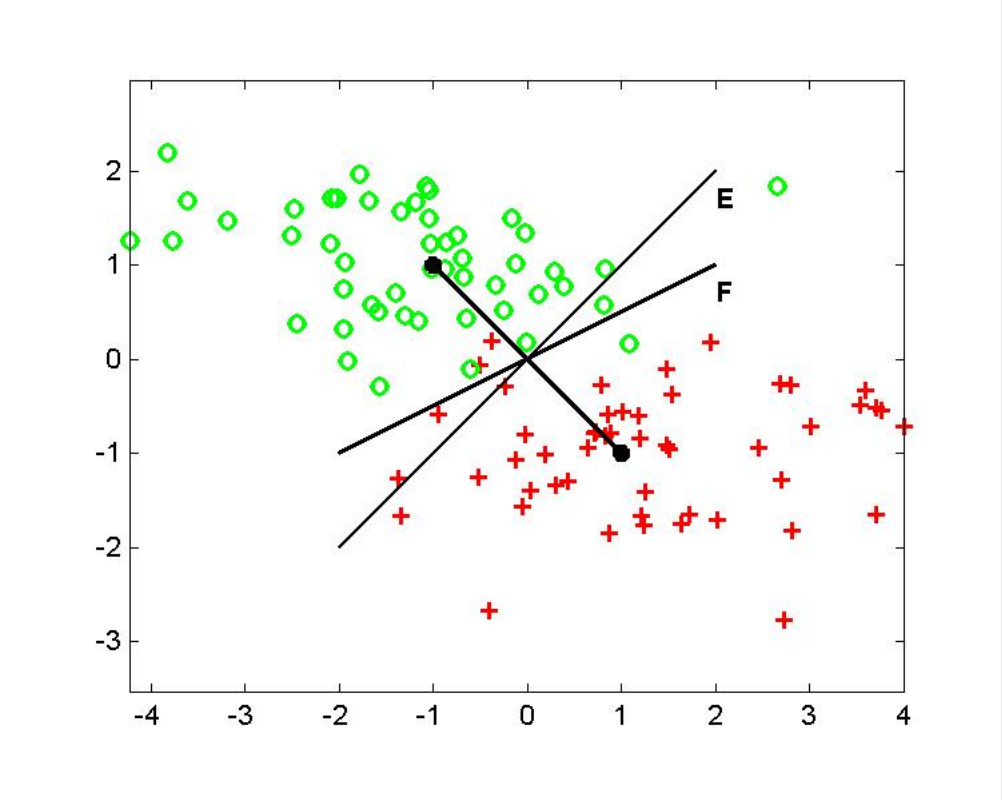
ir

$$c = -(\overline{\mathbf{M}}_1 + \overline{\mathbf{X}}M_2)'\mathbf{V}^F/2,$$

o empirinė kovariacijos matrica apskaičiuojama pagal formulę

$$\Sigma = \frac{\Sigma_{i=1}^{2}\Sigma_{j=1}^{N_i}(\mathbf{X}_j^i - \overline{\mathbf{X}}_i)(\mathbf{X}_j^i - \overline{\mathbf{X}}_i)'}{N_1 + N_2 - 2}$$

Jei grupių požymių vektoriai pasiskirstę pagal daugiamatį Gauso dėsnį su ta pačia kovariacine matrica ir skirtingais vidurkiais, tai



2 paveikslėlis: Dviejų grupių taškų atpažinimas artimiausiojo vidurkio ir Fišerio tiesiniais klasifikatoriais

standartinis Fišerio klasifikatorius yra optimalus. Naudojant vektorių $\mathbf{V}^F$ ir konstantą c. Fišerio tiesinis klasifikatorius išreiškiamas formule

$$h(\mathbf{X}) = \mathbf{X}'\mathbf{V}^F + c.$$

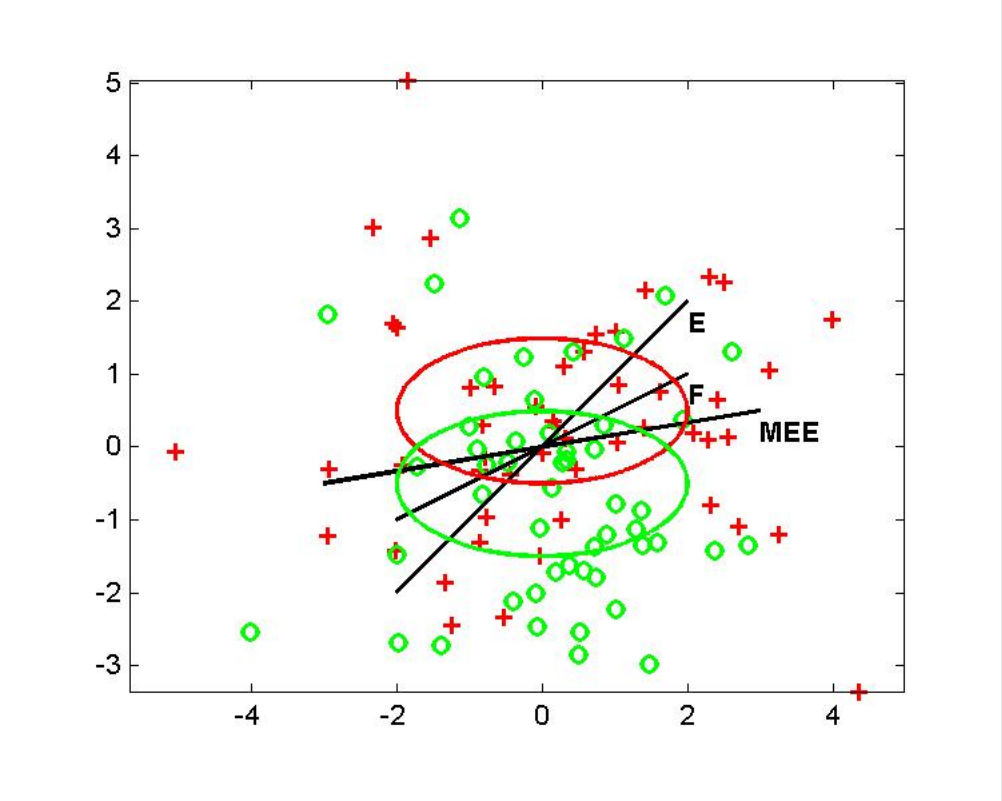
Ir vienu ir kitu atveju išvesdami klasifikavimo taisyklę mes darėme prielaidą apie duomenų pasiskirstymą, įvertindavome pasiskirstymo parametrus ir po to gaudavome klasifikavimo taisyklę. Silpna vieta čia yra daroma prielaida apie parametrinį pasiskirstymo dėsnį. Daugelis autorių teigia, kad vietoje prielaidos apie pasiskirstymo dėsnį geriau fiksuoti kokiai klasei turi priklausyti klasifikavimo taisyklė ir toje klasėje ieškoti optimaliai atskiriančio grupes klasifikatoriaus. Pavyzdžiui, galima fiksuoti, kad mūsų klasifikatoriaus funkcija tiesiškai priklausys nuo požymių vektoriaus ir tarpe visų tiesinių išraiškų ieškosime skiriamosios tiesės geriausiai skiriančios apmokymo duomenimis. Panašios klasifikavimo paradigmos laikosi dirbtinių neuroninių

tinklų pagrindu sukurti klasifikatoriai ir atraminių vektorių klasifikatoriai.

Susipažinsime su MEE (angl. *minimum empirical error*) klasifikatoriumi, kuris priklauso tiesinių klasifikatorių aibei ir randamas tiesiogiai minimizuojant klasifikavimo paklaidą.

Žemiau pateiktame brėžinyje palygintos klasifikavimo tiesės gautos taikant Euklidinį atstumą (E), Fišerio (F) ir minimalios empirinės paklaidos (MEE) metodą. Klasifikavimui pateiktos dvi plokštumos taškų grupės, kurios skiriasi vidurkiais: $\mu_1 = -\mu_2 = 0.5$. Prie abiejų grupių taškai pasiskirstę pagal normalųjį dėsnį su ta pačia kovariacine matrica, kurios kvadratinės formos vieną lygio liniją iliustruoja elipsė.

Iš brėžinio matyti, kad tiesiogiai duomenų imčiai adaptuotas tiesinis klasifikatorius (MEE) geriausiai atlieka skirtingų grupių taškų atpažinimą. Tačiau nebūtinai šis klasifikatorius yra geriausias. Neu-



3 paveikslėlis: Dviejų grupių taškų atpažinimas artimiausiojo vidurkio (E), Fišerio (F) ir optimizuojančiu pateiktos imties atpažinimą (MEE) tiesinio klasifikatoriaus.

roniniuose tinkluose susiduriama su reiškiniu, vadinamu tinklo *pertreniravimu, persimokymu* (angl. *overtraining*. Jo esmė yra per didelė modelio ar (ir) klasifikatoriaus parametrų adaptacija prie pateiktos mokymo duomenų specifikos. *Pertreniravimas* reiškia, kad klasifikatoriaus mokymo eigoje buvo pereitas etapas, kuriame einamosios parametrų reikšmės būtų davusios geresnius testavimo duomenims rezultatus. Su pertreniravimo reiškiniu galima kovoti panaudojant verifikavimo duomenis. Mokymo procesas nutraukiamas, kai verifikacijos rezultatai pradeda blogėti, nežiūrint į tai, kad mokymo duomenims klasifikavimas toliau optimizuojant parametrus gerėja.

1.2. Bajeso klasifikatorius

Bajeso klasifikatoriaus vardas kilęs iš Bajeso teoremos, kuri nurodo tokį sąryšį tarp sąlyginių tikimybių:

$$P(Y|X) = \frac{P(X|Y)\,P(Y)}{P(X)}. \tag{1}$$

Tikimybių teorijoje X ir Y yra bet kokie įvykiai. Tačiau klasifikavimo atveju atsiranda tam tikra asimetrija tarp X ir Y, nes čia bet kokia X realizacija laikoma duomenimis, o Y - klase. Dėl šios asimetrijos naudojamos tokios sąvokos:

- $P(Y)$ apriorinė (žinoma prieš bandymą) klasės Y tikimybė.
- $P(Y|X)$ sąlyginė klasės Y tikimybė, kai žinomi duomenys X.
- $P(X|Y)$ sąlyginė duomenų X tikimybė, kai žinoma, kad jie yra klasės Y. Ši tikimybė vadinama *tikėtinumu*.
- $P(X)$ apriorinė duomenų X tikimybė.

Klasifikuojant $P(X)$ tikimybė vaidina normuojančio daliklio vaidmenį ir neįtakoja klasifikavimo rezultatų. Susipažinsime su Bajeso klasifikatoriumi pritaikydami jį atpažinti MNIST duomenų bazės rašytinius skaitmenis. X duomenimis laikysime skaitmenų

28×28 binarizuotus vaizdus. Kadangi pradiniai MNIST vaizdai sudaryti iš pilkumo lygmenų reikšmių, reikia apibrėžti binarizacijos slenkstį. Mes paimsime trivialų slenkstį $treshold = 127.5$, t.y. laikysime, kad binarizuoto vaizdo reikšmė kokioje nors 28×28 matricos pozicijoje lygi 0, jei pilkumo lygmuo toje pozicijoje neviršija 127.5 ir lygi 1 priešingu atveju. Y elementai sudaryti iš dešimties skaitmenų, t.y. galimos Y reikšmės yra $\{0, 1, 2, \cdots, 9\}$. MNIST duomenų bazėje yra 60000 mokymo duomenų ir 10000 testavimo duomenų. Todėl galime laikyti, kad turime 60000 X ir Y egzempliorių

$$\{X_i, Y_i\}_{i=1}^{60000},$$

kuriuos galime naudoti Bajeso klasifikatoriaus apmokymui. Atpažįstant nežinomą skaičių mes turėsime 28×28 binarinę matricą X ir turėsime šiai matricai priskirti žymę Y, t.y. pasakyti koks skaitmuo pavaizduotas matricoje. Mūsų klasifikatoriaus tikslumui įvertinti

naudosime 10000 testavimo duomenų (skaitmenų paveiksliukų), kurių žymes žinome, tačiau šių duomenų nenaudosime kuriant klasifikavimo taisyklę. Visų galimų skirtingų paveiksliukų yra $2^{28\times 28} = 2^{784}$. Tai yra didžiulis skaičius ir turima 60000 skaitmenų mokymo aibė sudaro tik menkutę 2^{784} dalį. Tiksliau ta dalis lygi $\frac{60000}{2^{784}} = 5.910^{-232}$. Tai yra tipinė klasifikavimo uždavinių problema - mokymo duomenų, lyginant su visų galimų skirtingų duomenų X skaičiumi, yra labai nedaug. Bet koks klasifikavimo algoritmas (taisyklė) stengiasi apibendrinti turimus mokymo duomenis ir algoritmo kokybei įvertinti naudojami nauji nesantys mokymo duomenų imtyje duomenys. Klaidų procentas, kurį daro klasifikatorius atpažindamas nežinomus duomenis, vadinamas *apibendrinimo* klaida. Kad įvertinti klasifikatoriaus apibendrinimo klaidą, reikia tiksliai žinoti duomenų skirstinį ir visų galimų vaizdų klasifikavimo bei tikrąsias reikšmes. Tačiau praktiškai tiksliai apskaičiuoti apibendrinimo

klaidos neįmanoma, nes nežinomas nei tikslus duomenų skirstinys nei visų galimų duomenų tikslios žymės ($y_j \in Y$), nes priešingu atveju nereikėtų kurti klasifikatoriaus. Todėl praktiškai apibendrinimo paklaida įvertinama panaudojant testavimui skirtus duomenis. Didžiausia bėda čia, kad labai dažnai tobulinant klasifikavimo algoritmą jo kokybė daug kartų įvertinama panaudojant testavimo duomenis ir todėl galutinio algoritmo klasifikavimo apibendrinimo paklaida būna įvertinama per daug optimistiškai, t. y. algoritmo kūrimo metu parenkant optimalius klasifikatoriaus parametrus būna netiesiogiai atsižvelgiama į tarpinius algoritmo testavimo rezultatus.

Kad išspręsti šią problemą, rekomenduojama naudoti *kryžminio patikrinimo* (angl. *cross validation* metodą. Pagal šį metodą turima mokymo duomenų aibė suskaidoma į panašaus dydžio poaibius ir mokinimui skiriami visi poaibiai, išskyrus vieną, kuris naudojamas testavimui. Testavimui skirtas poaibis yra keičiamas ir, jei

leidžia skaičiavimo laiko resursai, pereina visas galimus mokymo imties suskaidymo poaibius. Dėl šios savybės metodas ir yra vadinamas kryžminio patikrinimo. Algoritmo parametrai yra optimizuojami kryžminės patikros poaibiams ir galutinė algoritmo klasifikavimo apibendrinimo paklaida įvertinama vieną kartą algoritmo kūrimo pabaigoje paskaičiuojant kiek galutinė algoritmo versija daro klaidų atpažindama testinius duomenis.

Kadangi mokymo duomenų visada trūksta, gana dažnai naudojamas *naivusis* Bajeso klasifikatorius. Naivusis klasifikatorius daro prielaidą, kad duomenų visos komponentės yra nepriklausomos, t. y.

$$P(X|Y) = P(\{x_1, x_2, ..., x_{784}\}|Y) = P(x_1|Y)P(x_2|Y)\dots P(x_{784}|Y). \tag{2}$$

Naiviojo Bajeso klasifikatoriaus prielaidą (2) užrašėme MNIST duomenų atveju. Epitetas "naivusis" pabrėžia, kad realiai tokia

sąlyga tiksliai niekada netenkinama. Tačiau praktiškai naivusis Bajeso klasifikatorius dažnai duoda neblogus rezultatus. Viena pagrindinių to priežasčių yra kas pavienių komponenčių skirstiniai gali būti patikimai įvertinti naudojant net ir nedidelias mokymo duomenų imtis. Mūsų atveju kiekviena duomenų komponentė gali įgyti tik dvi reikšmes: 0 arba 1. Tai dar labiau supaprastina skirstinio įvertį, nes mums reikia įvertinti tik

$$p_i^y = P(x_i = 1|y),\ i = 1, 2, \ldots, 784,\ y = 0, 1, \ldots, 9,$$

tikimybes, o likusias $q_i^y = P(x_i = 0|y)$ rasime iš sąryšio $p_i^y + q_i^y = 1$. p_i^y tikimybes įvertinsime panaudodami mokymo duomenis. Viso turime 60000 pavyzdžių. o atsižvelgus į skaitmenų reikmes, gauname tokius mokymo duomenis atskiriems skaitmenims:

y	0	1	2	3	4
	5	6	7	8	9
#	5923	6742	5958	6131	5842
	5421	5918	6265	5851	5949
$P(Y=y)$	$\frac{5923}{60000}$	$\frac{6742}{60000}$	$\frac{5958}{60000}$	$\frac{6131}{60000}$	$\frac{5842}{60000}$
	$\frac{5421}{60000}$	$\frac{5918}{60000}$	$\frac{6265}{60000}$	$\frac{5851}{60000}$	$\frac{5949}{60000}$

Pateikti lentelėje duomenys naudojami įvertinti apriorinę skaitmenų tikimybę $P(Y = y)$. Lentelės apatinėje eilutėje pateiktos įverčio reikšmės.

Kad įvertinti fiksuotai pozicijai i ir skaitmeniui y tikimybę $p_i^y = P(x_i = 1|y)$, pakanka suskaičiuoti kiek iš 60000 paveikslėlių turi žymę y ir turintiems žymę y suskaičiuoti vienetukų skaičių pozicijoje i. Žemiau pateiktoje lentelėje nurodyti vienetukų kiekiai pozicijose $i = 1, 289, 293, 297, 401, 405, 409, 513, 517, 521, 784$. Kraštinės pozicijos i reikšmės atitinka paveikslėlio kairįjį viršutinį ($i = 1$) ir dešinįjį apatinį ($i = 784$) kampą. Iš 1 lentelės duomenų matyti, kad visuose

mokymui skirtuose paveiksliukuose visi skaitmenys kraštinėse pozicijose turėjo tik juodas spalvas, todėl šiose pozicijose bendras vienetukų kiekis lygus 0. Vadinasi kraštinės pozicijos nesuteikia jokios naudingos informacijos skaitmenų klasifikacijai. Vidurinės i reikšmės parinktos taip, kad jos atitiktų 28×28 paveikslėlio $(14 \pm 4, 14 \pm 4)$ pozicijas. Šiose pozicijose skirtingiems skaitmenims vienetukų kiekis reikšmingai skiriasi ir jos naudingos klasifikacijai. Padalinę vienetukų kiekį iš skaitmens mokymo imties dydžio gausime tikimybes p_i^y. Pavyzdžiui

$$p_1^0 = 0,\ p_{289}^0 = \frac{2301}{5923},\ p_{405}^0 = \frac{519}{5923},\ p_{521}^0 = \frac{485}{5923};$$
$$p_1^1 = 0,\ p_{289}^1 = \frac{23}{6742},\ p_{405}^1 = \frac{1590}{6742},\ p_{521}^1 = \frac{4071}{6742};$$
$$p_1^9 = 0,\ p_{289}^9 = \frac{2162}{5949},\ p_{405}^9 = \frac{2760}{5949},\ p_{521}^9 = \frac{3627}{5949}.$$

Naivusis Bajeso klasifikatorius naudoja visas p_i^y ir $q_i^y = 1 - p_i^y$ tikimybes. Tarkime turime klasifikacijai pateiktą paveiksliuką $X =$

y/i	1	289	293	297	401	405	409	513	517	521	784
0	0	2301	3484	1253	4041	519	146	3788	485	2125	0
1	0	23	728	4187	16	1590	1617	109	4071	772	0
2	0	586	613	2199	623	1929	3395	3050	3991	3522	0
3	0	384	1048	3872	703	3790	3843	541	277	2002	0
4	0	1872	1981	1626	4035	2591	4628	823	1401	3343	0
5	0	1295	2918	1229	1556	3064	2291	759	532	2052	0
6	0	996	3325	427	2325	2463	3831	2606	3701	2977	0
7	0	2958	3200	3592	730	180	4113	122	1186	3660	0
8	0	1527	2694	1746	543	4276	4126	1524	3060	2558	0
9	0	2162	2345	2118	2950	2760	4958	388	982	3627	0

1 lentelė: Mokymo imties vienetukų kiekis nurodytose paveikslėlių pozicijose

$\{x_1, x_2, \ldots, x_{784}$. Tuomet tikimybė , kad šis paveiksliukas bus skaitmens y vaizdas pagal naiviojo Bajeso modelio algoritmą bus apskaičiuojama formule

$$P(Y = y|X) = \frac{P(X|Y = y)\, P(Y = y)}{P(X)} = \frac{P(Y = y) \prod_{i=1}^{784} (x_i p_i^y + (1 - x_i) q_i^y)}{P(X)}.$$

Vardiklio, t. y. tikimybės $P(X)$ mes apskaičiuoti negalime. Tačiau klasifikacijai jis nėra būtinas, nes pagal naiviojo Bajeso klasifikatorių skaitmuo y parenkamas taip, kad $P(Y = y|X)$ būtų maksimalus, o tam pakanka žinoti tik dešimties skaitiklių reikšmes, kurioms apskaičiuoti visi duomenys žinomi. Tarkime

$$x_1 = 0, \ldots, x_{289} = 0, \ldots, x_{293} = 0, \ldots, x_{297} = 1, \ldots,$$

$$x_{401} = 1, \ldots, x_{405} = 1, \ldots, x_{409} = 0, \ldots,$$

$$x_{513} = 1, \ldots, x_{517} = 0, \ldots, x_{521} = 1, \ldots, x_{784} = 0.$$

Tuomet

$$\begin{aligned} P(Y = 0|X) \;=\; & \tfrac{5923}{60000}(1 - \tfrac{0}{5923}) \cdots (1 - \tfrac{2301}{5923}) \cdots (1 - \tfrac{3484}{5923}) \cdots \\ & \tfrac{1253}{5923} \cdots \tfrac{4041}{5923} \cdots \tfrac{519}{5923}(1 - \tfrac{146}{5923}) \cdots \tfrac{3788}{5923} \cdots \\ & (1 - \tfrac{485}{5923}) \cdots \tfrac{2125}{5923} \cdots (1 - \tfrac{0}{5923})/P(X) \;, \end{aligned}$$

$$
\begin{aligned}
P(Y=1|X) &= \tfrac{6742}{60000}(1-\tfrac{0}{6742})\cdots(1-\tfrac{23}{6742})\cdots(1-\tfrac{728}{6742})\cdots \\
&\quad \tfrac{4187}{6742}\cdots\tfrac{16}{6742}\cdots\tfrac{1590}{6742}(1-\tfrac{1617}{6742})\cdots\tfrac{109}{6742}\cdots \\
&\quad (1-\tfrac{485}{4071})\cdots\tfrac{772}{6742}\cdots(1-\tfrac{0}{6742})/P(X)\ , \\
&\quad \cdots \\
P(Y=2|X) &= \tfrac{5949}{60000}(1-\tfrac{0}{5949})\cdots(1-\tfrac{2162}{5949})\cdots(1-\tfrac{2345}{5949})\cdots \\
&\quad \tfrac{2118}{5949}\cdots\tfrac{2950}{5949}\cdots\tfrac{2760}{5949}(1-\tfrac{4958}{5949})\cdots\tfrac{3788}{388}\cdots \\
&\quad (1-\tfrac{982}{5949})\cdots\tfrac{2125}{3627}\cdots(1-\tfrac{0}{5949})/P(X)\ .
\end{aligned}
$$

Čia daugtaškiai žymi praleistu sandaugos dėmenis, kurių reikšmių neturime 1 lentelėje. Kad išrinkti maksimalią tikimybę, pakanka apskaičiuoti tik skaitiklius, nes visų trupmenų vardiklis tas pats (nors ir yra nežinomas). Skaičiuojant praktiškai patartina skaičiuoti ne skaitiklio sandaugas, o sandaugos dėmenų logaritmų sumas, kad išvengti labai mažų skaičių aritmetikos. Skaičiuojant pagal naiviojo Bajeso taisyklę tikimybes turėsime problemų, kai kokia nors tikimybė lygi nuliui. Tuomet tokios tikimybės logaritmas neapibrėžtas ir gausime,

kad tokios žymės tikimybė lygi nuliui. Tai nelabai logiška, ypač kai mokymo duomenų nedaug, vienas iš 784 taškelių gali nulemti visos sandaugos reikšmę. Todėl rekomenduojama, jei kokio nors požymio x_i tikimybė lygi nuliui arba 1 priskirti ϵ ar $1-\epsilon$, kur ϵ mažas teigiamas skaičius, kurio reikšmė pasirenkama eksperimentiniu būdu maksimizuojant teisingai klasifikuojamų duomenų kiekį.

Klasifikavimas pagal naiviojo Bajeso taisyklę yra labai spartus. Iš tikrųjų, jei visos $P(Y = y)$ ir p_i^y tikimybės jau apskaičiuotos arba nuskaitytos iš failo, tai klasifikuojant vieną skaitmenį pakanka apskaičiuoti 784×10 sumas ir iš gautų dešimties skaičių išrinkti maksimumą ir klasifikacijos rezultatui pateikti maksimumą atitinkančių žymę. Klasifikatoriaus sukūrimas taip pat labai spartus - užtenka prabėgti vieną kartą visus mokymo duomenis, sukaupti tikimybėms apskaičiuoti reikalingą statistiką ir, turint vienetukų ir nuliukų skaitiklius, apskaičiuoti $P(Y = y)$ ir p_i^y tikimybes.

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 28 of 54

Go Back

Full Screen

Close

Quit

Naivusis Bajeso klasifikatorius (kaip ir kiti) negarantuoja, kad visi mokymo imties duomenys yra klasifikuojami teisingai. Todėl žemiau pateikiamos lentelės apie klasifikavimo rezultatus mokymo ir testavimo imtims atskirai.

- Mokymo imties dydis: 60000
- Teisingai klasifikuota (atpažinta): 50384 arba 83.9733 %
- Neteisingai klasifikuota skaitmenų: 9616 arba 16.0267 %

- Atskirų skaitmenų klasifikavimas:

0	1	2	3	4	5	6	7	8	9	$<--$
5371	1	50	15	21	99	166	1	153	46	0
0	6446	59	16	2	84	36	15	68	16	1
77	69	4919	145	68	16	277	90	256	41	2
44	81	344	4966	8	151	60	81	214	182	3
12	20	48	4	4469	36	116	11	106	1020	4
109	36	106	569	112	3885	137	15	206	246	5
66	74	136	2	48	207	5318	0	59	8	6
84	141	89	33	191	4	6	5355	72	290	7
57	89	104	209	60	202	80	8	4718	324	8
59	63	64	66	384	77	10	129	160	4937	9

- Naiviojo Bajeso modelio sukūrimo laikas: 2.25 sek.
- Mokymo duomenų klasifikavimo laikas: 48.69 sek.

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 30 of 54

Go Back

Full Screen

Close

Quit

- Testavimo imties dydis: 10000
- Teisingai klasifikuota (atpažinta): 8454 arba 84.54 %
- Neteisingai klasifikuota skaitmenų: 1546 arba 15.46 %

- Atskirų skaitmenų klasifikavimas:

0	1	2	3	4	5	6	7	8	9	$<--$
913	0	3	3	0	10	21	2	22	6	0
0	1095	12	1	0	7	8	2	10	0	1
14	14	833	36	7	1	45	18	53	11	2
4	3	54	845	0	26	7	21	28	22	3
2	3	7	1	740	7	21	1	21	179	4
13	4	10	96	22	653	26	2	31	35	5
18	6	19	1	14	30	855	0	13	2	6
8	28	33	1	15	1	1	871	18	52	7
10	6	12	41	11	30	11	6	791	56	8
13	9	8	10	60	16	1	10	24	858	9

- Testavimo duomenų klasifikavimo laikas: 8.1 sek.

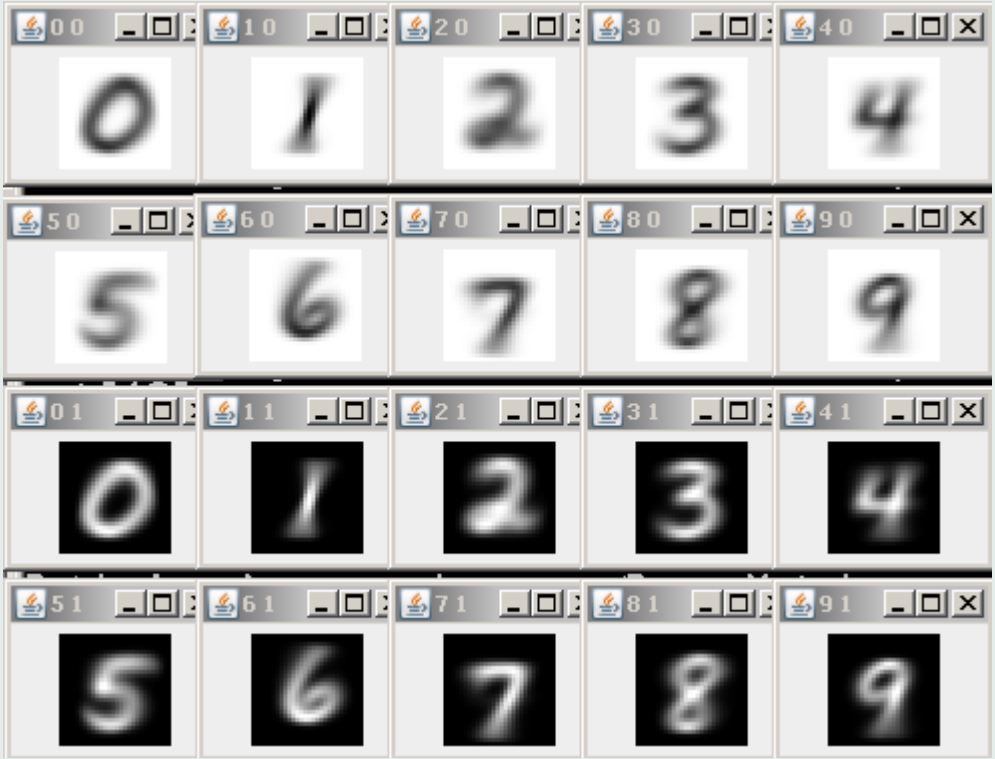
Gauta naiviojo Bajeso klasifikatoriaus kokybė yra gana gera, nes jei klasifikavimas būtų atsitiktinis, gautumėme tik apie 10 % teisingai klasifikuotų skaitmenų. Lyginant mokymo ir testavimo duomenų teisingai klasifikuotų skaitmenų dalį matome, kad testuojami duomenys klasifikuojami net šiek tiek geriau - tai byloja, kad klasifikatorius "nepersimokė", t.y. per daug nesiadaptavo prie mokymo imties specifinių paveikslėlių. 4 paveikslėlis iliustruoja tankio funkciją įvertintą pagal naiviojo Bajeso modelį MNIST mokymo duomenims. Matyti, kad reikšmės $x_i = 1$ yra labiausiai tikėtinos ten kur dažniausiai rašomas skaitmuo. Taip atsitinka dėl to, kad visi MNIST paveiksliukai yra standartinio dydžio ir centruoti. Kadangi testavimui skirti duomenys parinkti iš tos pačios NIST įrašų duomenų bazės, labiausiai tikėtina, kad skaitmens $x_i = 1$ pozicijos bus nelabai nutolusios nuo mokymo imties tankio $P(x_i = 1)$ maksimumų ir todėl tikimybė įvertinta pagal naiviojo Bajeso taisyklę bus didžiausia tuo atveju, kai žymės y

reikšmė sutaps su tikrąja skaitmens reikšme.

1.3. MNIST skaitmenų klasifikavimo pagerinimas

1.3.1. Klasifikacijos pagerinimas išlyginant skaitmenų nuožulnumą

Kadangi atpažįstami skaitmenys įrašyti ranka, jie būna įvairiai pakreipti. Pavyzdžiui rašant vienetuką, jį galima užrašyti tiesia linija, kuri eina iš viršaus tiesiai į apačią, arba ta linija gali eiti nuožulniai - dešiniau viršuje, kairiau apačioje, arba retkarčiais atvirkščiai, kairiau viršuje ir dešiniau apačioje. Kaupiant tokių duomenų Bajeso statistiką gausime mažiau koncentruotus nuliukų/vienetukų pasiskirstymus $28x28$ dažnių lentelėje. Sprenžiant šią problemą galimi bent du keliai. Pirmas įvertinti kokiu kampu pakrypęs parašytas skaitmuo ir pasukti pradinį paveikslėlį tokiu kampu, kad po posūkio gautumėme statmenai parašytą skaičių. Antras sprendimas įvertinti



4 paveikslėlis: Naiviojo Bajeso MNIST duomenų modelio tankio funkcijos vizualizacija. Šviesesni taškai žymi didesnį tankį ir atvirkščiai, tamsesni mažesnį tankį. Viršutinėse dviejose eilutėse $P(x_i = 0)$ reikšmės, apatinėse - $P(x_i = 0)$.

skaitmens parašymo šlytį ir pritaikyti pradiniam vaizdui šlyties operaciją po kurios skaitmuo būtų parašytas statmenai. Antrasis būdas turi kai kurių privalumų lyginant su pirmuoju. Posūkio operacija perveda ir x ir y koordinates į slankaus kablelio reikšmes ir todėl sukant tenka atlikti aproksimacijas ir abscisių ir ordinačių kryptimis. Šlyties atveju keičiasi naujo vaizdo tik x koordinatės, todėl aproksimacija bus atliekama tik abscisės kryptimi. Šlyties operacijos parametrui surasti gaunamos paprastesnės lygtys nei posūkio atveju, todėl galima tikėtis, kad skaitmens nuožulnumo atstatymas naudojant šlyties operaciją duos geresnius skaitmenų atpažinimo rezultatus.

Kad įvertinti šlyties operacijos parametrą, pradžioje formaliai apibrėžkime šlyties operaciją.

1 apibrėžimas. *Sakysime, kad vaizdas* $v(x, y)$ *yra vaizdo* $u(x, y)$

šlytis, jei

$$v(x, y) = u(x - \alpha(y - y_o), y).$$

Čia α *yra šlyties parametras, o* y_o *vaizdo aukščio pusė (mūsų atveju* $y_o = \frac{28}{2} = 14$ *).*

Vaizdo statmenumas gali būti įvertinas panaudojant tokius vaizdo momentus:

$$\begin{aligned}
\overline{u}_{XY} &= \sum_{x,y} u(x, y) * x * y, \\
\overline{u}_{YY} &= \sum_{x,y} u(x, y) * y * y, \\
\overline{u}_{X} &= \sum_{x,y} u(x, y) * x, \\
\overline{u}_{Y} &= \sum_{x,y} u(x, y) * y, \\
\overline{u}_{1} &= \sum_{x,y} u(x, y).
\end{aligned}$$

Jei kryžminė koreliacija lygi nuliui, laikysime, kad vaizdas yra statmenas. Prilyginę pakreipto vaizdo kyžminę koreliaciją nuliui, gausime tokią šlyties parametro išraišką:

$$\alpha = -\frac{\overline{u}_{XY}\overline{u}_1 - \overline{u}_X * \overline{u}_Y}{\overline{u}_{YY}\overline{u}_1 - \overline{u}_Y * \overline{u}_Y}. \tag{3}$$

.

Kadangi binarizuoto vaizdo $u(x, y)$ reikšmės yra tik dvi: 0 ir 1,

vaizdų momentai gali būti greitai apskaičiuoti pagal tokias formules:

$$\begin{aligned}
\overline{u}_{XY} &= \sum_{x,y:u(x,y)=1} x * y, \\
\overline{u}_{YY} &= \sum_{x,y:u(x,y)=1} y * y, \\
\overline{u}_{X} &= \sum_{x,y:u(x,y)=1} x, \\
\overline{u}_{Y} &= \sum_{x,y:u(x,y)=1} y, \\
\overline{u}_{1} &= \sum_{x,y:u(x,y)=1} 1.
\end{aligned}$$

5 paveikslėlis iliustruoja pirmuosius MNIS mokymo imties skaitmenis (kairiau) ir jų pakreiptus vaizdus (dešiniau). Atlikus MNIST skaitmenų vaizdų šlyties išlyginimą gaunamas apie 2% klasifikacijos pagerėjimas.

5 paveikslėlis: MNIST binarizuoti pradiniai duomenys (kairiau) ir su išlygintu nuožulnumu (dešiniau)

1.3.2. Klasifikacijos pagerinimas panaudojant papildomus požymių atributus

Skaitmens požymiais mes laikėme $784 = 28 \times 28$ binarizuoto vaizdo reikšmes. Pereitame skyrelyje mes įsisavinome techniką, kuri suteikia galimybę sumažinti požymių skaičių papildant juos atributais. Skaitmenų atveju galima įvertinti gradiento apibendrintus ekstremumus ir pažymėti ektremumo kryptį kvantuojant ją pasirinktu kvantų kiekiu. Tačiau tokie požymiai nepatogūs Bajeso klasifikatoriui, kadangi jų kiekis įvairiems skaitmenų vaizdams bus skirtingas. Kad išspręsti šią problemą, laikysime, kad skaitmens požymių skaičius sutampa su pradinio vaizdelio dydžiu (t.y. $784 = 28 \times 28$) laikydami, kad požymis yra neapibrėžtas, jei tame taške gradiento modulis nėra apibendrintas ekstremumas. Jei skaitmens vaizdo matricos taškas atitinka apibendrintą gradiento modulio ekstremumą, tokį tašką laikysime požymiu ir jo reikšmė sutaps su kvantuota krypties reikšme. Lai-

kysime, kad viso yra H kvantuotų krypčių reikšmių. Mūsų praktinių užduočių atveju $H = 8$, dabar mes parinksime H vertę maksimizuodami teisingai klasifikuojamų skaitmenų kiekį.

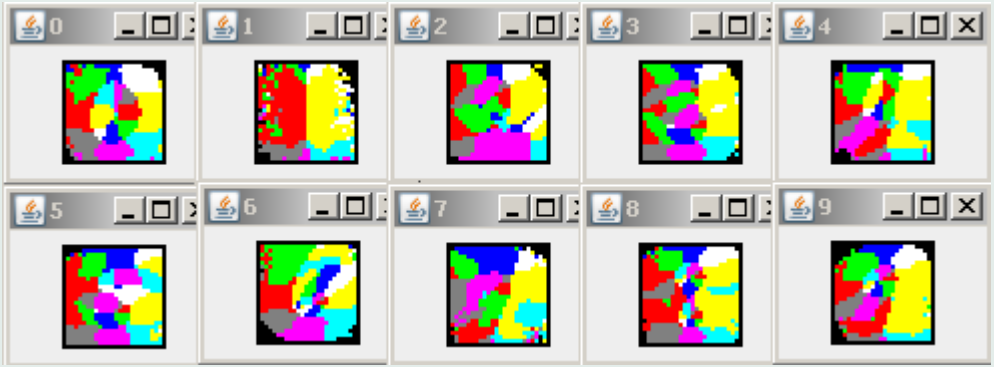
6 paveikslėlis iliustruoja pirmųjų dviejų MNIST mokymo skaitmenų (viso 20) požymius su spalvos atributu, kai $H = 8$. Taškeliai, kurie nėra gradiento modulio apibendrintieji ekstremumai, žymimi juodai. Sekantis fig:MNIST8spalvos paveikslėlis iliustruoja gradiento modulio dominuojančią kryptį kiekvieno skaitmens atveju. Šis paveikslėlis gaunamas tokiu būdu: surandami kiekvieno mokymo imties skaitmens požymiai su spalvos atributu, kiekvienam skaitmeniui atskirai surandama kiek kokiame taškelyje kartų pasikartojo kryptis ir gale iliustruojama ta kryptis, kuri dažniausiai pasitaikė duotajam skaitmeniui. Jei kokioje nors pozicijoje nei karto nebuvo apibendrintojo gradiento modulio maksimumo (tokios pozicijos būna vaizdo kraštuose), taškelis spalvinamas juodai. Tiesiogiai do-

6 paveikslėlis: Mokymo imties gradiento modulio dominuojančios kryptys. Krypčių kvantų kiekis $H = 8$

minuojantys kryptys nenaudojamos Bajeso tikėtinumų skaičiavimui, tačiau jas galima panaudoti, kad greitai įvertinti tikėtinumą iš viršaus. Jei greitai įvertintas kokiam nors skaitmeniui tikėtinumas mažas, gali atmesti hipotezę, kad tikrinamas skaičius yra duotasis skaitmuo ir pereiti prie sekančios hipotezės. Tačiau čia greito įverčio realizacijos detalių neaprašinėsime ir jo nenaudosime.

1.3.3. Klasifikacijos pagerinimas panaudojant požymių praplėtimą

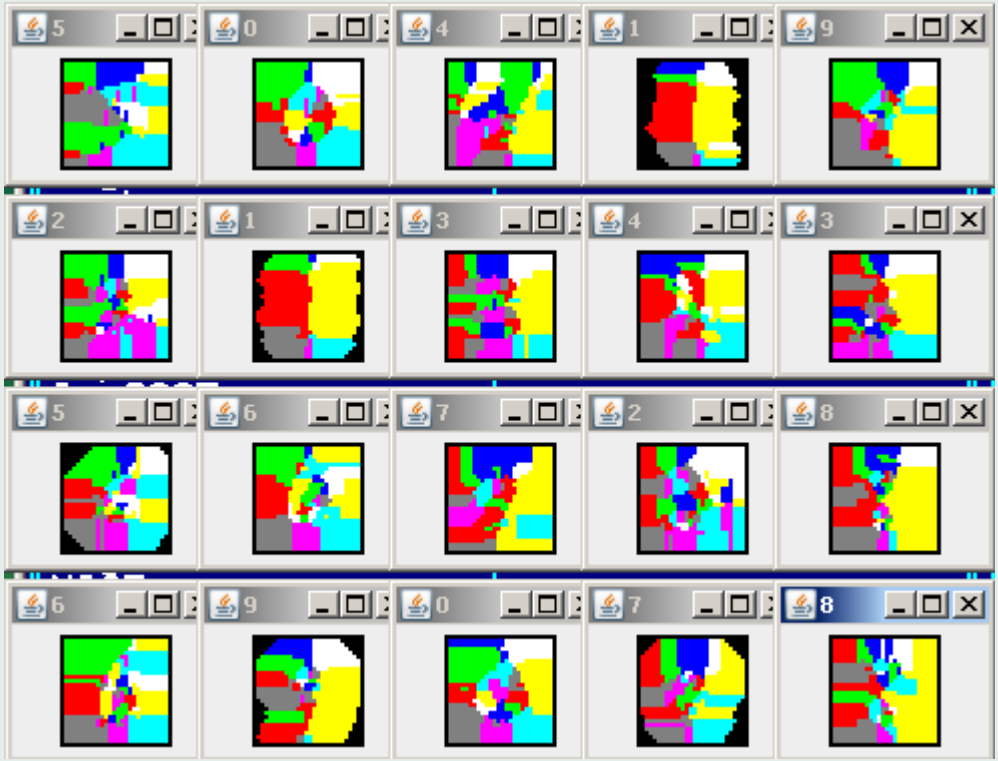
Kaip matyti iš 6 iliustracijos, didelė dalis paveikslėlio pozicijų neturi informacijos apie požymius, nes yra nuspalvinti juodai. Todėl kyla mintis praplėsti kiekvieno mokymo imties paveikslėlio požymių kiekį praplečiant juo bangos principu.



7 paveikslėlis: Mokymo imties gradiento modulio dominuojančios kryptys. Krypčių kvantų kiekis $H = 8$

1.3.4. Klasifikacijos pagerinimas atliekant mokymo duomenų vektorinį kvantavimą

Tradicinis Bajeso klasifikatorius laiko, kad vienos klasės duomenų požymiai turi tą patį skirstimį. Tačiau ši prielaida ne visada yra natūrali. Tarkime rašant skaitmenis vieni asmenys vienetuką rašo



8 paveikslėlis: Mokymo imties gradiento modulio dominuojančios kryptys. Krypčių kvantų kiekis $H = 8$

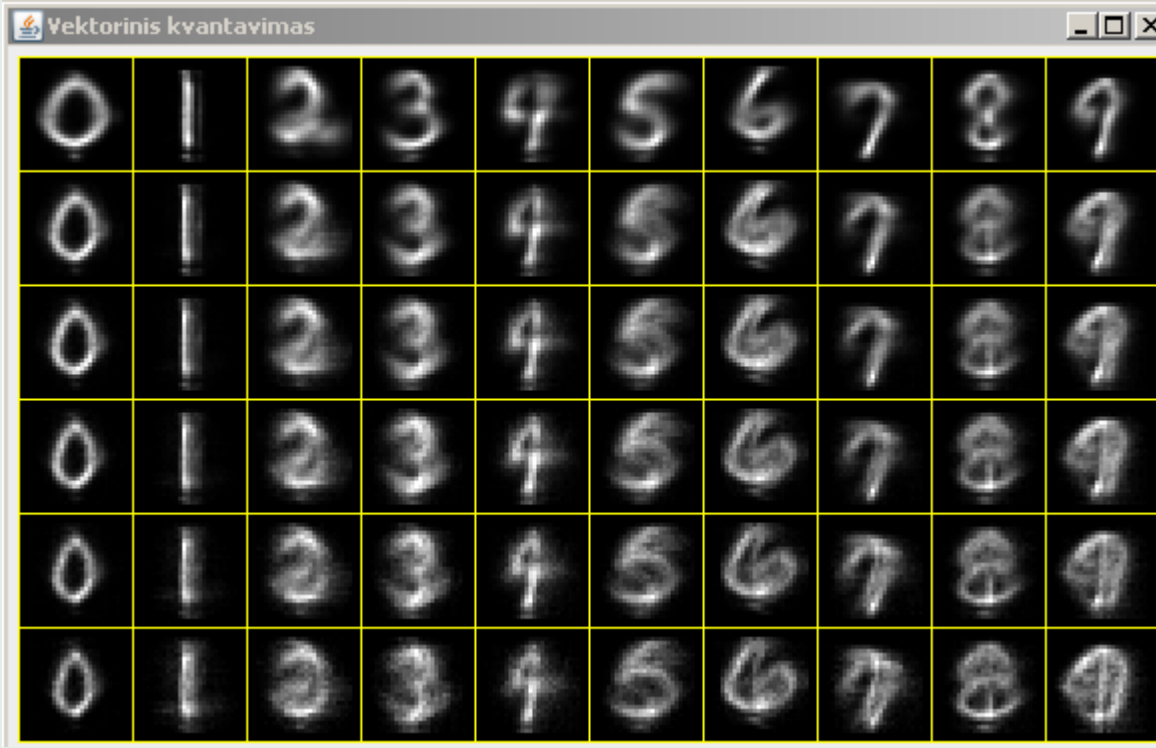
vienu brūkšneliu, kiti prie pagrindinio vertikalaus brūkšnelio viršuje iš kairės prideda mažą brūkšnelį. Septynetas gali būti parašytas panaudojant perbraukimą ir be jo. Nulis gali būti ir platus ir siauras ir taip toliau. Todėl kyla mintis įvertinti kiekvieno skaitmens ne vieną požymių skirstinį, o jų seriją. Gana sudėtingai numatyti visus galimus skaitmenų rašymo stilius ir įvertinti praktiškai kokiu stiliumi parašytas skaitmuo. Tam tikslui galima panaudoti Bajeso klasifikatorių. Idėja tokia:

- Fiksuotam skaitmeniui panaudojame visą turimą mokymo duomenų imtį, kad įvertinti skaitmens požymių skirstinį.
- Įvertiname skaitmens tikėtinumą visai turimai mokymo imčiai.
- Surūšiuojame tikėtinumus didėjimo tvarka.
- Pasirenkame tam tikrą skaitmenų dalį, kurių tikėtinumai yra artimi, ir iš jų suformuojame naują skaitmens požymių skirstinį.

Viena šios idėjos realizacijų:

- Pirmame etape naudojame $[q_0, q_1]$ dalį didžiausio tikėtinumo mokymo duomenų; $q_0 = 0$, $q_1 = 2/3$.
- Sekančiame etape naudojame $[q_0, q_1]$ dalį didžiausio tikėtinumo mokymo duomenų, kur q_0 ir q_1 apskaičiuoti pagal taisyklę $q_0 = q_1$, $q1 = q0 + (1 - q0)/3$.
- Antro punkto taisyklę kartojame penkis kartus imdami paskutinį kartą $q_1 = 1$.

Aprašyta procedūra artima vektoriniam kvantavimui. Vektorinis kvantavimas grupuoja duomenų požymių vektorius į grupes pagal požymių artumą. Mes tai ir atlikome, tik šiuo atveju vietoje kokios nors požymių artumo metrikos buvo panaudota tikėtinumo reikšmė įvertinta pagal Bajeso modelį.



9 paveikslėlis: Mokymo imties gradiento modulio dominuojančios kryptys. Krypčių kvantų kiekis $H = 8$

9 paveikslėlis iliustruoja skeletizuotų skaitmenų Bajeso vienetukų tikėtinumus; kuo taškelis baltesnis tuo labiau tikėtina, kad tame taškelyje bus skaitmens skeleto taškelis. Stulpeliai žymi skaitmens reikšmę (nuo nulio iki 9), o eilutės išsirinktos $[q_0, q_1]$ mokymo imties dalies skeleto taškų tikimybes (baltesni taškeliai žymi didesnes tikimybes). Pirmoje eilutėje labiausiai tikėtinos skeleto pozicijos atrodo ryškiausiai. Taip yra dėl to, kad dauguma asmenų panašiai piešia skaitmenis ir dauguma lemia skeletų pozicijas. Didėjant eilutės numeriui Bajeso tankio statistikai apskaičiuoti naudojami vis mažiau tikėtini skaitmenų užrašai, todėl šiose eilutėse skaitmenų užrašymas labiau varijuoja ir todėl skeleto taškų maksimalių tikėtinumų pozicijos labiau išplitusios. Atkreipkite dėmesį kaip keičiasi nulio maksimalaus tikėtinumo iliustracijos. Sprendžiant pagal gautus vaizdus, dažniausiai nulis rašomas platesnis ir rečiau siauresnis. Tą patį galima pasakyti apie ketvertuko ir aštuoneto užrašymą, tačiau penketai ir

šešetai atvirkščiau dažniau rašomi glaustai ir rečiau plačiai. Taip pat, sprendžiant pagal skeletų išsibarstymą, devynetai ir dvejetai rašomi gana variabiliai ir todėl turėtų kilti problemų juos atpažįstant pagal naiviojo Bajeso metodą.

Atpažinimas turint keletą tikėtinumų variantų vienam skaitmeniui atliekamas maksimizuojant tiriamo skaitmens tikėtinumą. Pavyzdžiui jei tikėtinumui įvertinti naudojame 9 iliustracijoje pavaizduotus skeleto taškų tankius, tai duotajam testavimo skaitmens skeletui įvertiname tikėtinumą visiems 6×10 tankiams ir išsirenkame iš šešiasdešimties didžiausio tikėtinumo reikšmę ir, žinodami iš kokio mokymo skaitmens buvo gautas maksimalų tikėtinumą atitinkantis tankis, priskiriame klasifikacijos žymę. Apskaičiavę kiekvienam skaitmeniui maksimalius tankius (maksimizuodami kiekvieno stulpelio tikėtinumą), gausime įvertį apie kiekvieno skaitmens tikėtinumą.

1.3.5. Pagerinto naiviojo Bajeso MNIST klasifikacijos testavimo skaitmenų rezultatai

Kiekvienas pasiūlytas rašytinių skaitmenų atpažinimo pagerinimo būdas pagerina atpažinimo kokybę vienu-dviem procentais.

Apačioje pateikti galutiniai MNIST testavimo imties rašytinių skaitmenų atpažinimo rezultatai. Pagerintas naiviojo Bajeso klasifikatorius daro apie 2 procentus (1.96%) klaidų. Tokie skaitmenų klasifikavimo rezultatai gauti glodinant su eksponentiniu filtru, kurio $\sigma = 0.5$, kvantuotų krypčių kiekis $H = 180$.

0	1	2	3	4	5	6	7	8	9	#	%	z
974	0	1	0	0	0	3	1	1	0	980	99.39%	0
0	1127	4	1	0	0	2	0	1	0	1135	99.3%	1
1	0	1021	2	0	0	2	2	4	0	1032	98.93%	2
0	0	2	999	0	4	0	3	2	0	1010	98.91%	3
1	0	2	0	967	0	6	0	1	5	982	98.47%	4
1	0	0	5	1	882	2	0	1	0	892	98.88%	5
5	3	1	0	3	2	941	0	3	0	958	98.23%	6
1	3	21	3	2	0	0	992	2	4	1028	96.5%	7
2	1	2	6	5	3	1	0	940	14	974	96.51%	8
1	5	2	7	18	2	1	8	4	961	1009	95.24%	9
986	1139	1056	1023	996	893	958	1006	959	984			
98,8	98,9	96,7	97,7	97,1	98,8	98,2	98,6	98,0	97,7		98.04%	

Literatūra

[1] Bill Green, http://www.pages.drexel.edu/ weg22/edge.html

[2] Kálmán Palágyi, Vengrija, http://www.inf.u-szeged.hu/ palagyi/skel/skel.html

[3] K. Stukas, J. Janauskas, Š. Gruodis, M. Brašiškis, http://www.mif.vu.lt/ bastys/academic/ATE/skaiciai/skaiciu_atp.htm#Praktinis

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 54 of 54

Go Back

Full Screen

Close

Quit

[4] D. Rutovitz, Pattern Recognition, J. Roy. Statist. Soc., vol. 129, pp. 504-530, 1966.

[5] Feng Zhao and Xiaoou Tang, CISST02 International Conference, http://mmlab.ie.cuhk.edu.hk/2002/CISST02_Fingerprint.pdf, (Lokali kopija http://mif.vu.lt/ bastys/academic/ATE/pirshtai/CISST02_Fingerprint.pdf)

[6] T. Y. Zhang, C. Y. Suen, A fast parallel algorithm for thinning digital patterns, Communications of the ACM, v.27 n.3, p.236-239, March 1984.
Realizacija Java kalba: http://www.mif.vu.lt/atpazinimas/skaiciai/skelet/Zh