Week 2, Lecture 3 - Fitting & Regression Redux, Regularization

Aaron Meyer

Outline

- Administrative Issues
- Fitting Regularization
  - Lasso
  - Ridge regression
  - Elastic net
- Some Examples

Based on slides from Rob Tibshirani.

The Bias-Variance Tradeoff

The Bias-Variance Tradeoff

Estimating β

- As usual, we assume the model:

$$y = f(\mathbf{z}) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)$$

- In regression analysis, our major goal is to come up with some good regression function

$$\hat{f}(\mathbf{z}) = \mathbf{z}^\mathsf{T} \hat{\beta}$$

- So far, we've been dealing with $\hat{\beta}^{ls}$, or the least squares solution:
  - $\hat{\beta}^{ls}$ has well known properties (e.g., Gauss-Markov, ML)
- But can we do better?

Choosing a good regression function

- Suppose we have an estimator

$$\hat{f}(\mathbf{z}) = \mathbf{z}^\mathsf{T}\hat{\beta}$$

- To see if this is a good candidate, we can ask ourselves two questions:
    1. Is $\hat{\beta}$ close to the true β?
    2. Will $\hat{f}(\mathbf{z})$ fit future observations well?
- These might have very different outcomes!!

Is $\hat{\beta}$ close to the true β?

- To answer this question, we might consider the **mean squared error** of our estimate $\hat{\boldsymbol{\beta}}$:
  - i.e., consider squared distance of $\hat{\boldsymbol{\beta}}$ to the true $\boldsymbol{\beta}$:

$$MSE(\hat{\boldsymbol{\beta}}) = \mathbb{E}\left[\left\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right\|^2\right] = \mathbb{E}[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\intercal(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})]$$

- **Example:** In least squares (LS), we now that:

$$\mathbb{E}[(\hat{\boldsymbol{\beta}}^{ls} - \boldsymbol{\beta})^\intercal(\hat{\boldsymbol{\beta}}^{ls} - \boldsymbol{\beta})] = \sigma^2 \mathrm{tr}[(\mathbf{Z}^T\mathbf{Z})^{-1}]$$

Will $\hat{f}(z)$ fit future observations well?

- Just because $\hat{f}(z)$ fits our data well, this doesn't mean that it will be a good fit to new data
- In fact, suppose that we take new measurements y_i' at the same $\mathbf{z}_i$'s:

$$(\mathbf{z}_1, \mathbf{y}_1'), (\mathbf{z}_2, \mathbf{y}_2'), ..., (\mathbf{z}_n, \mathbf{y}_n')$$

- So if $\hat{f}(\cdot)$ is a good model, then $\hat{f}(\mathbf{z}_i)$ should also be close to the new target y_i'
- This is the notion of **prediction error** (PE)

Prediction error and the bias-variance tradeoff

- So good estimators should, on average have, small prediction errors
- Let's consider the PE at a particular target point $\mathbf{z}_0$:
  - $PE(\mathbf{z}_0) = \sigma_\epsilon^2 + Bias^2(\hat{f}(\mathbf{z}_0)) + Var(\hat{f}(\mathbf{z}_0))$
  - Not going to derive, but comes directly from previous definitions
- Such a decomposition is known as the **bias-variance tradeoff**
  - As model becomes more complex (more terms included), local structure/curvature can be picked up
  - But coefficient estimates suffer from high variance as more terms are included in the model
- So introducing a little bias in our estimate for β might lead to a substantial decrease in variance, and hence to a substantial decrease in PE

Depicting the bias-variance tradeoff

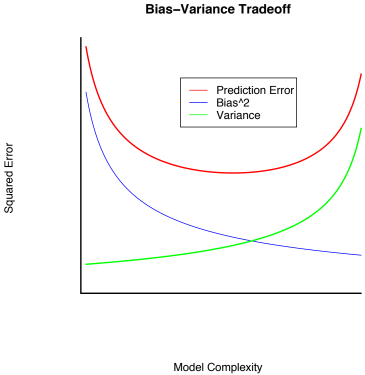


Figure: A graph depicting the bias-variance tradeoff.

Ridge Regression

Ridge Regression

Ridge regression as regularization

- If the β_j's are unconstrained…
  - They can explode
  - And hence are susceptible to very high variance
- To control variance, we might **regularize** the coefficients
  - i.e., Might control how large the coefficients grow
- Might impose the ridge constraint (both equivalent):
  - minimize $\sum_{i=1}^{n}(y_i - \boldsymbol{\beta}^\intercal \mathbf{z}_i)^2 \text{ s.t.} \sum_{j=1}^{p} \beta_j^2 \leq t$
  - minimize $(y - \mathbf{Z}\boldsymbol{\beta})^\intercal (y - \mathbf{Z}\boldsymbol{\beta}) \text{ s.t.} \sum_{j=1}^{p} \beta_j^2 \leq t$
- By convention (very important!):
  - **Z** is assumed to be standardized (mean 0, unit variance)
  - **y** is assumed to be centered

Ridge regression: l_2-penalty

- Can write the ridge constraint as the following **penalized** residual sum of squares (PRSS):

$$\begin{aligned} PRSS(\boldsymbol{\beta})_{\ell_2} &= \sum_{i=1}^{n}(y_i - \mathbf{z}_i^\top \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \\ &= (\mathbf{y} - \mathbf{Z}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}) + \lambda ||\boldsymbol{\beta}||_2^2 \end{aligned}$$

- Its solution may have smaller average PE than $\hat{\boldsymbol{\beta}}^{ls}$
- $PRSS(\boldsymbol{\beta})_{l_2}$ is convex, and hence has a unique solution
- Taking derivatives, we obtain:

$$\frac{\delta PRSS(\beta)_{l_2}}{\delta \beta} = -2\mathbf{Z}^T(y - \mathbf{Z}\beta) + 2\lambda\beta$$

The ridge solutions

- The solution to $PRSS(\hat{\beta})_{l2}$ is now seen to be:

$$\hat{\beta}_{\lambda}^{ridge} = (\mathbf{Z}^{\intercal}\mathbf{Z} + \lambda\mathbf{I}_p)^{-1}\mathbf{Z}^{\intercal}\mathbf{y}$$

  - Remember that **Z** is standardized
  - **y** is centered
- Solution is indexed by the tuning parameter λ (more on this later)
- Inclusion of λ makes problem non-singular even if $\mathbf{Z}^{\intercal}\mathbf{Z}$ is not invertible
  - This was the original motivation for ridge regression (Hoerl and Kennard, 1970)

Tuning parameter λ

- Notice that the solution is indexed by the parameter λ
  - So for each λ, we have a solution
  - Hence, the λ's trace out a path of solutions (see next page)
- λ is the shrinkage parameter
  - λ controls the size of the coefficients
  - λ controls amount of **regularization**
  - As λ decreases, we obtain the least squares solutions
  - As λ increases, we have $\hat{\beta}^{ridge}_{\lambda=0} = 0$ (intercept-only model)

Ridge coefficient paths

- The λ's trace out a set of ridge solutions, as illustrated below

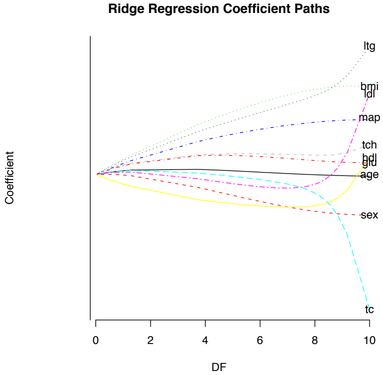


Figure: Ridge coefficient path for the diabetes data set found in the `lars` library in R.

Choosing λ

- Need disciplined way of selecting λ
- That is, we need to "tune" the value of λ
- In their original paper, Hoerl and Kennard introduced **ridge traces**:
  - Plot the components of $\hat{\beta}_{\lambda}^{ridge}$ against λ
  - Choose λ for which the coefficients are not rapidly changing and have "sensible" signs
  - No objective basis; heavily criticized by many
- Standard practice now is to use cross-validation (next lecture!)

Orthonormal **Z** in ridge regression

- If **Z** is orthonormal, then $\mathbf{Z}^T\mathbf{Z} = \mathbf{I}_p$, then a couple of closed form properties exist
- Let $\hat{\beta}^{ls}$ denote the LS solution for our orthonormal **Z**; then

$$\hat{\boldsymbol{\beta}}_{\lambda}^{ridge} = \frac{1}{1+\lambda}\hat{\boldsymbol{\beta}}^{ls}$$

- The optimal choice of λ minimizing the expected prediction error is:

$$\lambda^* = \frac{p\sigma^2}{\sum_{j=1} p\beta_j^2},$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)$ is the true coefficient vector

Smoother matrices and effective degrees of freedom

- A **smoother matrix S** is a linear operator satisfying:

$$\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$$

  - Smoothers put the "hats" on **y**
  - So the fits are a linear combination of the y_i's, $i = 1, \ldots, n$
- **Example:** In ordinary least squares, recall the hat matrix

$$\mathbf{H} = \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T$$

  - For rank(**Z**) = p, we know that tr(**Z**) = p, which is how many degrees of freedom are used in the model
- By analogy, define the **effective degrees of freedom** (or effective number of parameters) for a smoother to be:

$$df(\mathbf{S}) = tr(\mathbf{S})$$

Degrees of freedom for ridge regression

- In ridge regression, the fits are given by:

$$\hat{\mathbf{y}} = \mathbf{Z}(\mathbf{Z}^\intercal\mathbf{Z} + \lambda\mathbf{I}_p)^{-1}\mathbf{Z}^T\mathbf{y}$$

- So the smoother or "hat" matrix in ridge takes the form:

$$\mathbf{S}_\lambda = \mathbf{Z}(\mathbf{Z}^\intercal\mathbf{Z} + \lambda\mathbf{I}_p)^{-1}\mathbf{Z}^T$$

- So the effective degrees of freedom in ridge regression are given by:

$$df(\lambda) = tr(S_\lambda) = tr[\mathbf{Z}(\mathbf{Z}^\intercal\mathbf{Z} + \lambda\mathbf{I}_p)^{-1}\mathbf{Z}^T] = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}$$

  - Note that $df(\lambda)$ is monotone decreasing in λ
  - Question: What happens when $\lambda = 0$?

How do we choose λ?

- We need a disciplined way of choosing λ
- Obviously want to choose λ that minimizes the mean squared error
- Issue is part of the bigger problem of **model selection**

K-Fold Cross-Validation

- A common method to determine λ is K-fold cross-validation.
- **We will discuss this next lecture.**

Plot of CV errors and standard error bands

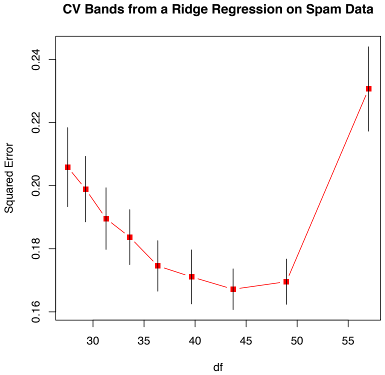


Figure: Cross validation errors from a ridge regression example on spam data.

The LASSO

The LASSO

The LASSO: l_1 penalty

- Tibshirani (*J of the Royal Stat Soc* 1996) introduced the **LASSO**: *least absolute shrinkage and selection operator*
- LASSO coefficients are the solutions to the l_1 optimization problem:

$$\text{minimize}\,(\mathbf{y}-\mathbf{Z}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{Z}\boldsymbol{\beta})\text{ s.t. }\sum_{j=1}^{p}\|\beta_j\|\leq t$$

- This is equivalent to loss function:

$$PRSS(\boldsymbol{\beta})_{l_1}=\sum_{i=1}^{n}(y_i-\mathbf{z}_i^T\boldsymbol{\beta})^2+\lambda\sum_{j=1}^{p}\|\beta_j\|$$

$$=(\mathbf{y}-\mathbf{Z}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{Z}\boldsymbol{\beta})+\lambda\|\boldsymbol{\beta}\|_1$$

λ (or t) as a tuning parameter

- Again, we have a tuning parameter λ that controls the amount of regularization
- One-to-one correspondence with the threshhold t:
  - recall the constraint:

$$\sum_{j=1}^{p} = \|\beta_j\| \leq t$$

  - Hence, have a "path" of solutions indexed by t
  - If $t_0 = \sum_{j=1}^{p} \left\|\hat{\beta}_j^{ls}\right\|$ (equivalently, $\lambda = 0$), we obtain no shrinkage (and hence obtain the LS solutions as our solution)
  - Often, the path of solutions is indexed by a fraction of shrinkage factor of t_0

Sparsity and exact zeros

- Often, we believe that many of the β_j's should be 0
- Hence, we seek a set of **sparse solutions**
- Large enough λ (or small enough t) will set some coefficients exactly equal to 0!
    - So the LASSO will perform model selection for us!

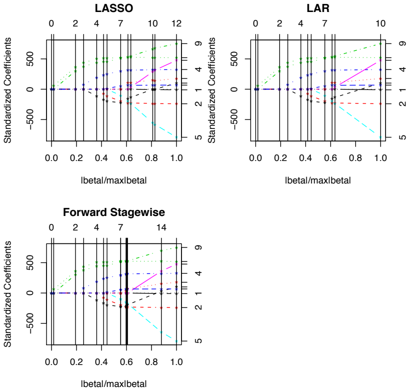
Computing the LASSO solution

- Unlike ridge regression, $\hat{\beta}_{\lambda}^{lasso}$ has no closed form λ
- Original implementation involves quadratic programming techniques from convex optimization
- But Efron et al. (Annals of Statistics 2004) proposed LARS (least angle regression), which computes the LASSO path efficiently
  - Interesting modification called is called forward stagewise
  - In many cases it is the same as the LASSO solution
  - Forward stagewise is easy to implement: http://www-stat.stanford.edu/~hastie/TALKS/nips2005.pdf

Forward stagewise algorithm

- As usual, assume **Z** is standardized and **y** is centered
- Choose a small ϵ. The forward-stagewise algorithm then proceeds as follows:
  1. Start with initial residual $\mathbf{r} = \mathbf{y}$, and $\beta_1 = \beta_2 = \ldots = \beta_p = 0$
  2. Find the predictor $\mathbf{Z}_j (j = 1, \ldots, p)$ most correlated with $\mathbf{r}$
  3. Update $\beta_j = \beta_j + \delta_j$, where $\delta_j = \epsilon \cdot \mathrm{sign}\langle \mathbf{r}, \mathbf{Z}_j \rangle = \epsilon \cdot \mathrm{sign}(\mathbf{Z}_j^T \mathbf{r})$
  4. Set $\mathbf{r} = \mathbf{r} - \delta_j \mathbf{Z}_j$
  5. Repeat from step 2 many times

The LASSO, LARS, and Forward Stagewise paths


LASSO

LAR

Forward Stagewise

Figure: Comparison of the LASSO, LARS, and Forward Stagewise coefficient paths for the diabetes data set.

Comparing LS, Ridge, and the LASSO

- Even though $\mathbf{Z}^T\mathbf{Z}$ may not be of full rank, both ridge regression and the LASSO admit solutions
- We have a problem when $p \gg n$ (more predictor variables than observations)
  - But both ridge regression and the LASSO have solutions
  - Regularization tends to reduce prediction error

More comments on variable selection

- Now suppose $p \gg n$
- Of course, we would like a parsimonious model (Occam's Razor)
- Ridge regression produces coefficient values for each of the p-variables
- But because of its l_1 penalty, the LASSO will set many of the variables exactly equal to 0!
  - That is, the LASSO produces **sparse solutions**
- So LASSO takes care of model selection for us
  - And we can even see when variables jump into the model by looking at the LASSO path

Variants

- Zou and Hastie (2005) propose the **elastic net**, which is a convex combination of ridge and the LASSO
  - Paper asserts that the elastic net can improve error over LASSO
  - Still produces sparse solutions

High-dimensional data and underdetermined systems

- In many modern data analysis problems, we have $p \gg n$
  - These comprise “high-dimensional” problems
- When fitting the model $y = \mathbf{z}^\intercal \beta$, we can have many solutions
  - i.e., our system is *underdetermined*
- Reasonable to suppose that most of the coefficients are exactly equal to 0

S-sparsity and Oracles

- Suppose that only S elements of β are non-zero
    - Candès and Tao call this S-sparsity
- Now suppose we had an "Oracle" that told us which components of the $\beta = (\beta_1, \beta_2, \ldots, \beta_p)$ are truly non-zero
- Let β^* be the least squares estimate of this "ideal" estimator:
    - So β^* is 0 in every component that β is 0
    - The non-zero elements of β^* are computed by regressing $\mathbf{y}$ on only the S important covariates

The Danzig Selector

- Candès and Tao developed the Dantzig selector $\hat{\beta}^{Dantzig}$:

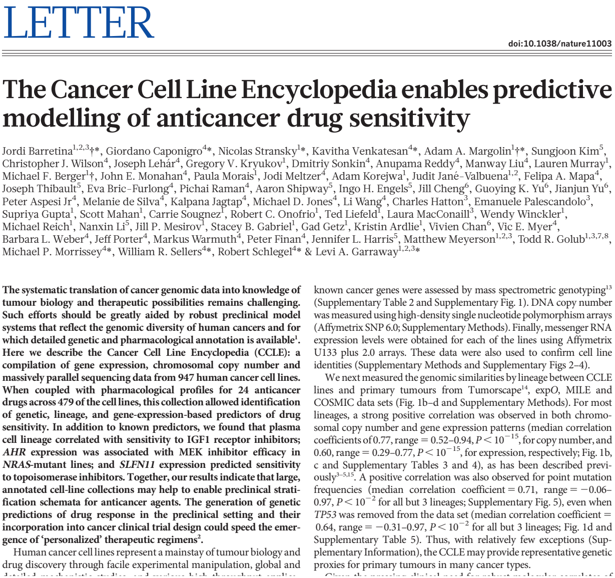
$$\min \|\beta\|_{l_1} \text{ s.t. } \left\|\mathbf{Z}_j^\intercal \mathbf{r}\right\|_{l_\infty} \leq (1 + t^{-1})\sqrt{2\log p} \cdot \sigma$$

  - Here, $\mathbf{r}$ is the residual vector and $t > 0$ is a scalar
- They showed that with high probability,

$$\left\|\hat{\beta}^{Dantzig} - \beta\right\|^2 = O(\log p)\,\mathbb{E}(\|\beta^* - \beta\|^2)$$

- So the Dantzig selector does comparably well as someone who was told which S variables to regress on

Example - Predicting Drug Response



LETTER

doi:10.1038/nature11003

The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity

Jordi Barretina[1,2,3]†*, Giordano Caponigro[4]*, Nicolas Stransky[1]*, Kavitha Venkatesan[4]*, Adam A. Margolin[1]†*, Sungjoon Kim[5], Christopher J. Wilson[4], Joseph Lehár[4], Gregory V. Kryukov[1], Dmitriy Sonkin[4], Anupama Reddy[4], Manway Liu[4], Lauren Murray[1], Michael F. Berger[1]†, John E. Monahan[4], Paula Morais[1], Jodi Meltzer[4], Adam Korejwa[1], Judit Jané-Valbuena[1,2], Felipa A. Mapa[4], Joseph Thibault[5], Eva Bric-Furlong[4], Pichai Raman[4], Aaron Shipway[5], Ingo H. Engels[5], Jill Cheng[6], Guoying K. Yu[6], Jianjun Yu[6], Peter Aspesi Jr[4], Melanie de Silva[4], Kalpana Jagtap[4], Michael D. Jones[4], Li Wang[4], Charles Hatton[3], Emanuele Palescandolo[3], Supriya Gupta[1], Scott Mahan[1], Carrie Sougnez[1], Robert C. Onofrio[1], Ted Liefeld[1], Laura MacConaill[3], Wendy Winckler[1], Michael Reich[1], Nanxin Li[5], Jill P. Mesirov[1], Stacey B. Gabriel[1], Gad Getz[1], Kristin Ardlie[1], Vivien Chan[6], Vic E. Myer[4], Barbara L. Weber[4], Jeff Porter[4], Markus Warmuth[4], Peter Finan[4], Jennifer L. Harris[5], Matthew Meyerson[1,2,3], Todd R. Golub[1,3,7,8], Michael P. Morrissey[4]*, William R. Sellers[4]*, Robert Schlegel[4]* & Levi A. Garraway[1,2,3]*

The systematic translation of cancer genomic data into knowledge of tumour biology and therapeutic possibilities remains challenging. Such efforts should be greatly aided by robust preclinical model systems that reflect the genomic diversity of human cancers and for which detailed genetic and pharmacological annotation is available[1]. Here we describe the Cancer Cell Line Encyclopedia (CCLE): a compilation of gene expression, chromosomal copy number and massively parallel sequencing data from 947 human cancer cell lines. When coupled with pharmacological profiles for 24 anticancer drugs across 479 of the cell lines, this collection allowed identification of genetic, lineage, and gene-expression-based predictors of drug sensitivity. In addition to known predictors, we found that plasma cell lineage correlated with sensitivity to IGF1 receptor inhibitors; *AHR* expression was associated with MEK inhibitor efficacy in *NRAS*-mutant lines; and *SLFN11* expression predicted sensitivity to topoisomerase inhibitors. Together, our results indicate that large, annotated cell-line collections may help to enable preclinical stratification schemata for anticancer agents. The generation of genetic predictions of drug response in the preclinical setting and their incorporation into cancer clinical trial design could speed the emergence of 'personalized' therapeutic regimens[2].

Human cancer cell lines represent a mainstay of tumour biology and drug discovery through facile experimental manipulation, global and

known cancer genes were assessed by mass spectrometric genotyping[13] (Supplementary Table 2 and Supplementary Fig. 1). DNA copy number was measured using high-density single nucleotide polymorphism arrays (Affymetrix SNP 6.0; Supplementary Methods). Finally, messenger RNA expression levels were obtained for each of the lines using Affymetrix U133 plus 2.0 arrays. These data were also used to confirm cell line identities (Supplementary Methods and Supplementary Figs 2–4).

We next measured the genomic similarities by lineage between CCLE lines and primary tumours from Tumorscape[14], exp*O*, MILE and COSMIC data sets (Fig. 1b–d and Supplementary Methods). For most lineages, a strong positive correlation was observed in both chromosomal copy number and gene expression patterns (median correlation coefficients of 0.77, range = 0.52–0.94, $P < 10^{-15}$, for copy number, and 0.60, range = 0.29–0.77, $P < 10^{-15}$, for expression, respectively; Fig. 1b, c and Supplementary Tables 3 and 4), as has been described previously[3–5,15]. A positive correlation was also observed for point mutation frequencies (median correlation coefficient = 0.71, range = −0.06–0.97, $P < 10^{-2}$ for all but 3 lineages; Supplementary Fig. 5), even when *TP53* was removed from the data set (median correlation coefficient = 0.64, range = −0.31–0.97, $P < 10^{-2}$ for all but 3 lineages; Fig. 1d and Supplementary Table 5). Thus, with relatively few exceptions (Supplementary Information), the CCLE may provide representative genetic proxies for primary tumours in many cancer types.

Example - Predicting Drug Response

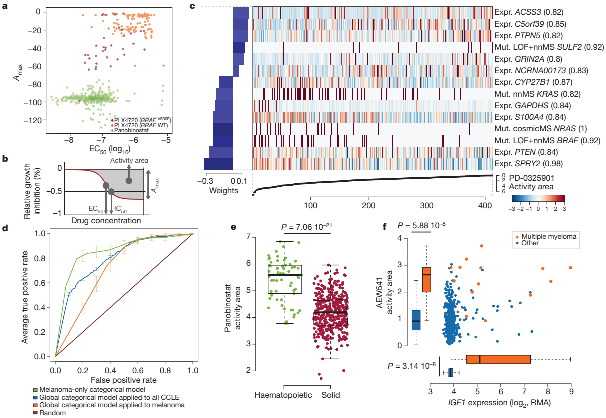


Figure 2 | Predictive modelling of pharmacological sensitivity using CCLE genomic data. **a**, **b**, Drug responses for panobinostat (green) and PLX4720 (orange/purple) represented by the high-concentration effect level (A_{max}) and transitional concentration (EC_{50}) for a sigmoidal fit to the response curve (**b**). **c**, Elastic net regression modelling of genomic features that predict sensitivity to PD-0325901. The bottom curve indicates drug response, measured as the area over the dose–response curve (activity area), for each cell line. The central heat map shows the CCLE features in the model (continuous *z*-score for expression and copy number, dark red for discrete mutation calls), across all cell lines (*x* axis). Bar plot (left): weight of the top predictive features for sensitivity (bottom) or insensitivity (top). Parentheses indicate features present in >80% of models after bootstrapping. LOF, loss of function mutation; nnMS, non-neutral missense mutation (Supplementary Methods). **d**, Specificity and sensitivity (receiver operating characteristic curves) of cross-validated categorical models predicting the response to a MEK inhibitor, PD-0325901 (activity area). Mean true positive rate and standard deviation ($n = 5$) are shown when models are built using all lines (global categorical model, in blue and orange), or within only melanoma lines (green). **e**, Activity area values for panobinostat between cell lines derived from haematopoietic ($n = 61$) and solid tumours ($n = 387$). The middle bar, median; box, inter-quartile range; bars extend to 1.5× the inter-quartile range. **f**, Distribution of activity area values for AEW541 relative to *IGF1* mRNA expression. Orange dots, multiple myeloma cell lines ($n = 14$); blue dots, cell lines from other tumour types ($n = 434$). Box-and-whisker plots show the activity area or mRNA expression distributions relative to each cell line type (line, median; box, inter-quartile range), with bars extending to 1.5× the inter-quartile range.

Example - Predicting Drug Response

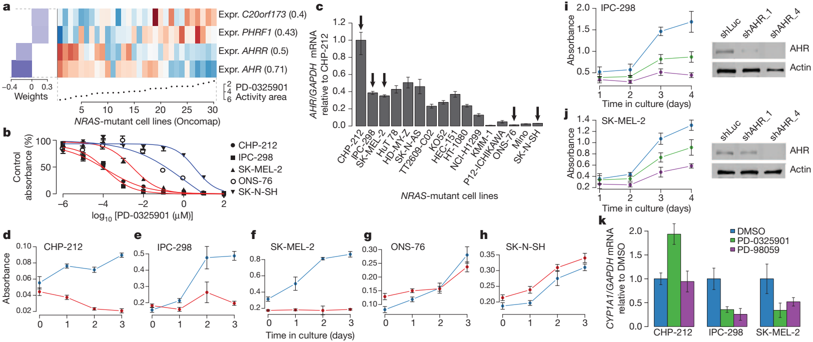


Figure 3 | *AHR* expression may denote a tumour dependency targeted by MEK inhibitors in *NRAS*-mutant cell lines. **a**, Predictive features for PD-0325901 sensitivity (using the 'varying baseline' activity area) in validated *NRAS*-mutant cell lines. **b**, Growth inhibition curves for *NRAS*-mutant cell lines expressing high (red) or low (blue) levels of *AHR* mRNA in the presence of the MEK inhibitor PD-0325901. **c**, Relative *AHR* mRNA expression across a panel of *NRAS*-mutant cell lines (arrows indicate cell lines where AHR dependency was analysed). **d**–**h**, Proliferation of *NRAS*-mutant cell lines displaying high (**d**–**f**) and low (**g**, **h**) *AHR* mRNA expression, after introduction of shRNAs against *AHR* (red lines) or luciferase (blue lines). **i**, Left: proliferation of IPC-298 cells (high AHR) after introduction of additional shRNAs against *AHR* (shAHR_1 and shAHR_4; green and purple lines, respectively) or luciferase (control shLuc; blue line). Right: corresponding immunoblot analysis of AHR protein. **j**, Equivalent studies as in **i** using SK-MEL-2 cells (high AHR). **k**, Endogenous *CYP1A1* mRNA expression in the neuroblastoma line CHP-212 or the melanoma lines IPC-298 and SK-MEL-2 after exposure to vehicle (blue) or MEK inhibitors (PD-0325901, green or PD-98059, purple). Error bars indicate standard deviation between replicates, with *n* = 12 (**b**), *n* = 3 (**c**), *n* = 6 (**d**–**k**).

Example - Predicting Drug Response

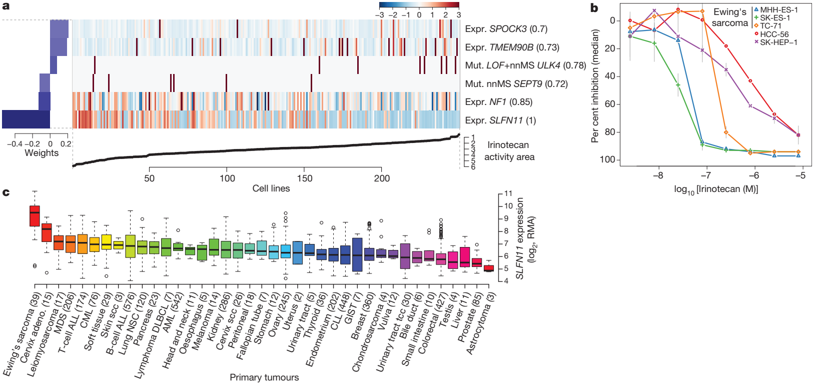


Figure 4 | Predicting sensitivity to topoisomerase I inhibitors. **a**, Elastic net regression analysis of genomic correlates of irinotecan sensitivity is shown for 250 cell lines. **b**, Dose–response curves for three Ewing's sarcoma cell lines (MSS-ES-1, SK-ES-1 and TC-71) and two control cell lines with low *SLFN11* expression (HCC-56 and SK-HEP-1). Grey vertical bars, standard deviation of the mean growth inhibition ($n = 2$). **c**, *SLFN11* expression across 4,103 primary tumours. Box-and-whisker plots show the distribution of mRNA expression for each subtype, ordered by the median *SLFN11* expression level (line), the inter-quartile range (box) and up to 1.5× the inter-quartile range (bars). Sample numbers (n) are indicated in parentheses.

Implementation

```
import numpy as np, matplotlib.pyplot as plt
from sklearn.metrics import r2_score
from sklearn.linear_model import Lasso, ElasticNet

# Generate some sparse data to play with
n_samples, n_features = 50, 200
X = np.random.randn(n_samples, n_features)
coef = 3 * np.random.randn(n_features)
inds = np.arange(n_features)
np.random.shuffle(inds)
coef[inds[10:]] = 0  # sparsify coef
y = np.dot(X, coef)

# add noise
y += 0.01 * np.random.normal(size=n_samples)

# Split data in train set and test set
n_samples = X.shape[0]
X_train, y_train = X[:n_samples // 2], y[:n_samples // 2]
X_test, y_test = X[n_samples // 2:], y[n_samples // 2:]
#
```

Implementation

```python
#...
# Split data in train set and test set
n_samples = X.shape[0]
X_train, y_train = X[:n_samples // 2], y[:n_samples // 2]
X_test, y_test = X[n_samples // 2:], y[n_samples // 2:]

# Lasso
alpha = 0.1
lasso = Lasso(alpha=alpha)

y_pred_lasso = lasso.fit(X_train, y_train).predict(X_test)
r2_score_lasso = r2_score(y_test, y_pred_lasso)

# ElasticNet
enet = ElasticNet(alpha=alpha, l1_ratio=0.7)

y_pred_enet = enet.fit(X_train, y_train).predict(X_test)
r2_score_enet = r2_score(y_test, y_pred_enet)
#...
```

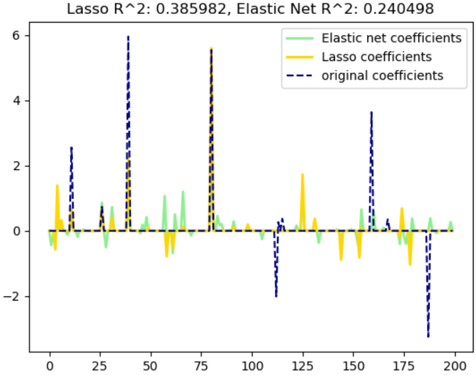
Implementation

```
#...
# ElasticNet
enet = ElasticNet(alpha=alpha, l1_ratio=0.7)

y_pred_enet = enet.fit(X_train, y_train).predict(X_test)
r2_score_enet = r2_score(y_test, y_pred_enet)

plt.plot(enet.coef_, color='lightgreen', linewidth=2,
         label='Elastic net coefficients')
plt.plot(lasso.coef_, color='gold', linewidth=2,
         label='Lasso coefficients')
plt.plot(coef, '--', color='navy', label='original coefficients'
plt.legend(loc='best')
plt.title("Lasso R^2: %f, Elastic Net R^2: %f"
          % (r2_score_lasso, r2_score_enet))
plt.show()
```

Implementation



Further Reading

- Computer Age Statistical Inference, Chapter 16
- sklearn: Generalized Linear Models
- Candès E. and Tao T. The Dantzig selector: statistical estimation when p is much larger than n.